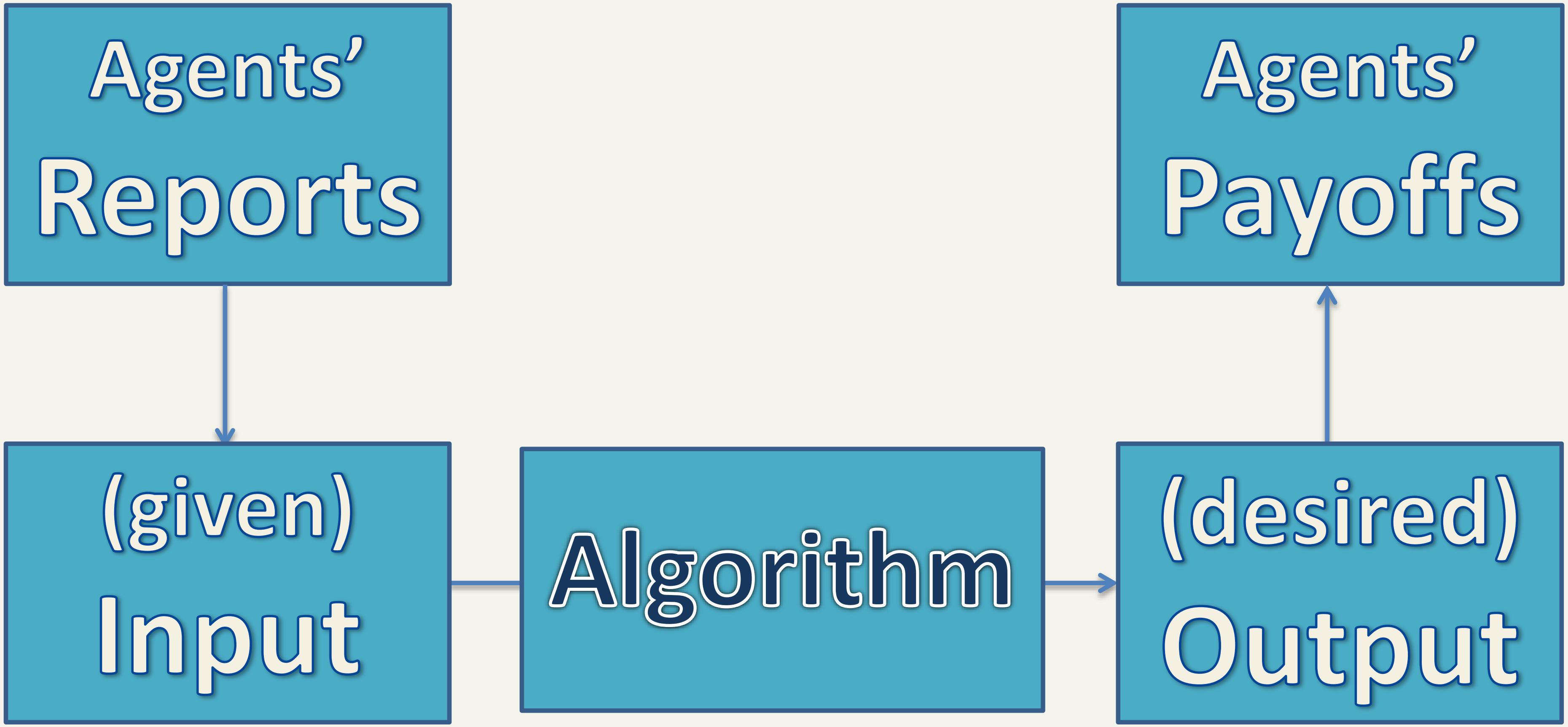
Mechanism Design for Learning Agents

Costis Daskalakis
(MIT)

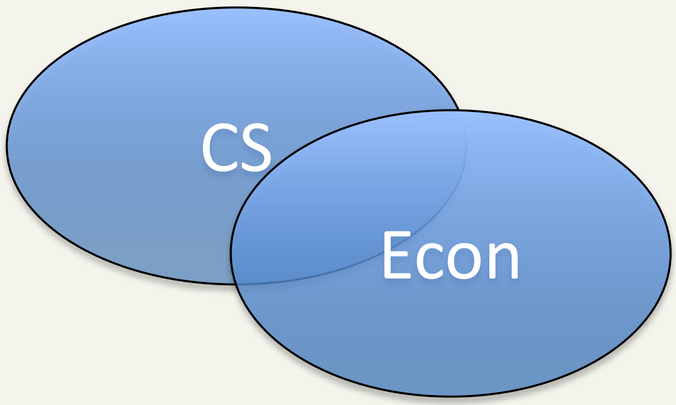
Vassilis Syrkanis
(Microsoft)

Algorithm Design

Algorithm Design in Practice



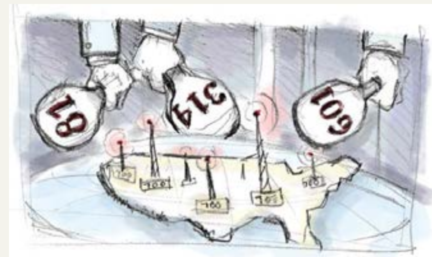
CS ∩ Econ Applications



CS ∩ Econ Applications

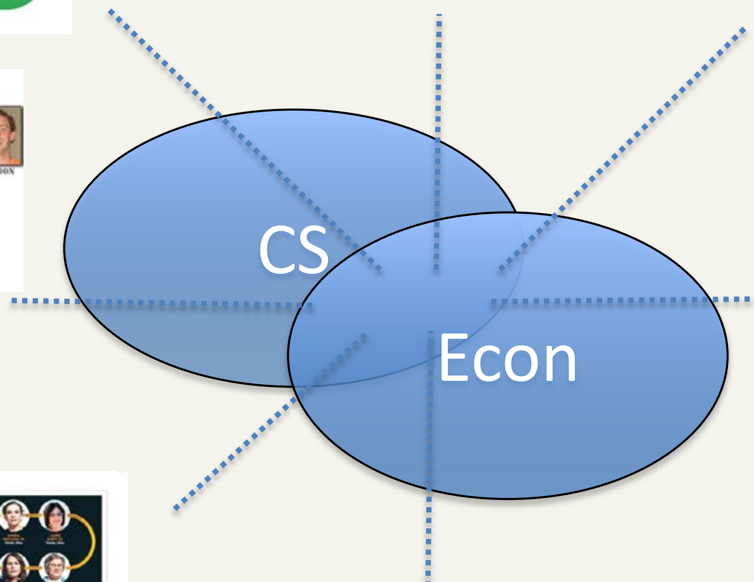
Online Markets & Advertising

Public Good Auctions



National security

game bots

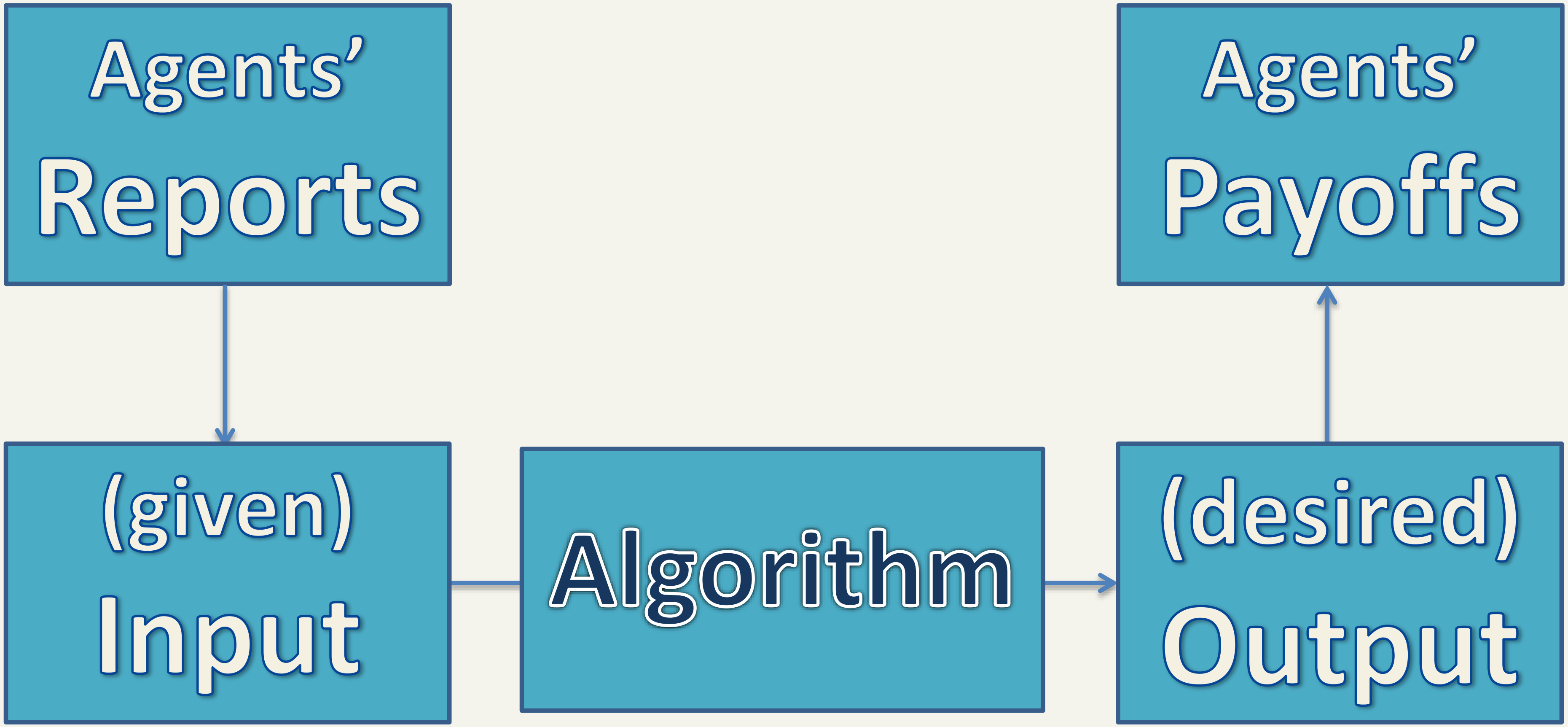


Sharing Economy

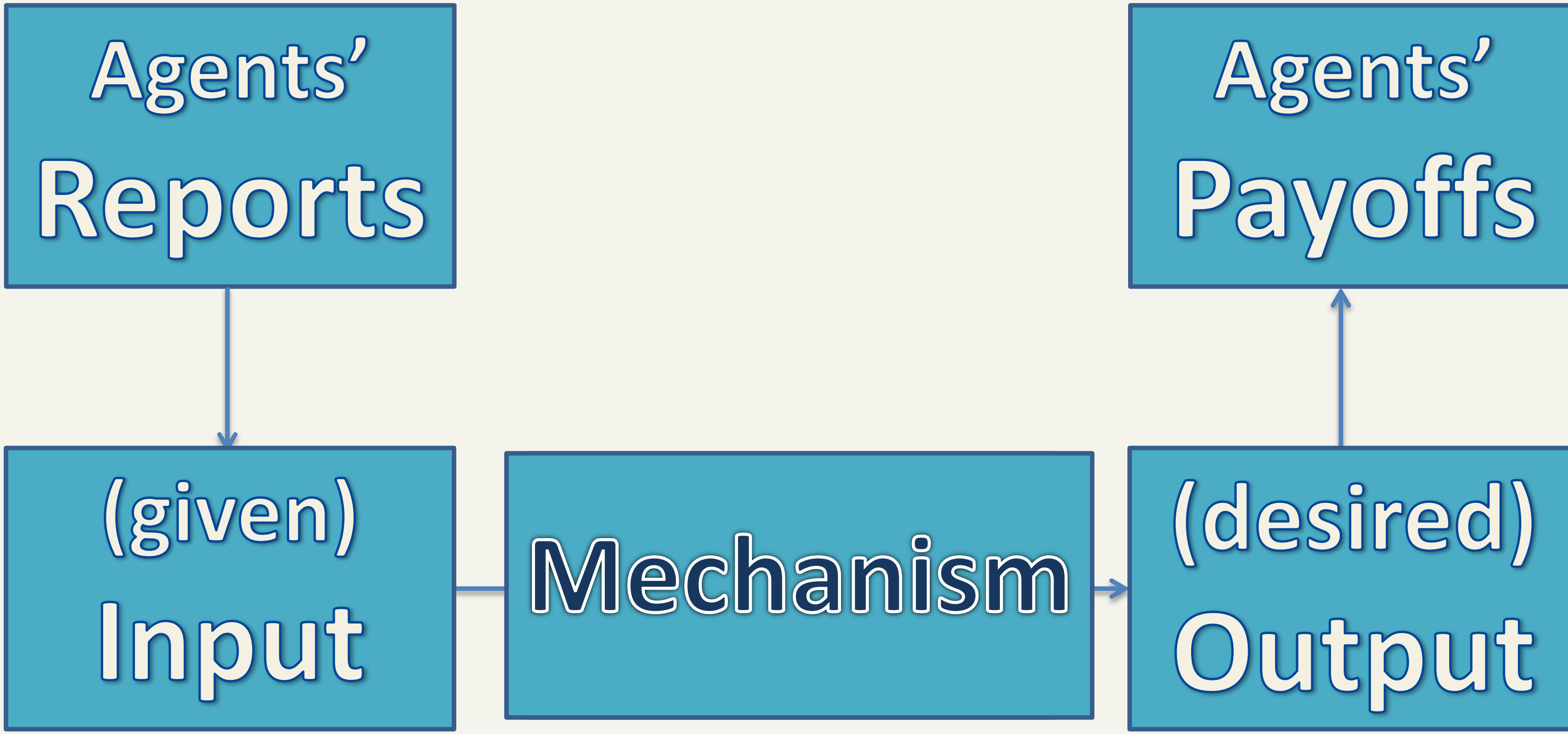
kidney exchanges

Crypto-currencies

Mechanism vs Algorithm Design



Mechanism vs Algorithm Design



E.g. Computing the Max

- **Input:** $x_1, x_2, \ldots, x_n$
- **Goal**: compute $\max(x_1, \ldots, x_n)$
- **Algorithm:** Trivial

E.g. Computing the Max

- **Input:** $x_1, x_2, \ldots, x_n$
- **Goal**: compute $\max(x_1,\ldots,x_n)$
- **Algorithm:** Trivial
- But what if inputs are strategic?
  - suppose input i has value x_i for being selected&algorithm doesn't know x_i
  - facing trivial algorithm, every input reports $+\infty$

E.g. Computing the Max

- **Input:** $x_1, x_2, \ldots, x_n$
- **Goal**: compute $\max(x_1, \ldots, x_n)$
- **Algorithm:** Trivial
- But what if inputs are strategic?
  - suppose input i has value x_i for being selected&algorithm doesn't know x_i
  - facing trivial algorithm, every input reports $+\infty$
- **A better Algorithm [Vickrey'61 ]:**
  - collect reported inputs: $b_1, \ldots, b_n$ (can't enforce $b_i = x_i$ a priori)
  - select $i^* = \arg\max b_i$
  - charge winner i^* the 2nd highest report: $\arg\max_{j \neq i^*} b_j$

E.g. Computing the Max

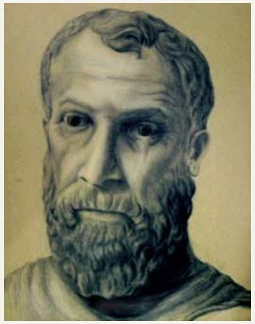
- **Input:** $x_1, x_2, \ldots, x_n$
- **Goal**: compute $\max(x_1,\ldots,x_n)$
- **Algorithm:** Trivial
- But what if inputs are strategic?
  - suppose input i has value x_i for being selected&algorithm doesn't know x_i
  - facing trivial algorithm, every input reports $+\infty$
- **A better Algorithm [Vickrey'61]:**
  - collect reported inputs: $b_1,\ldots, b_n$ (can't enforce $b_i=x_i$ a priori)
  - select $i^* = \arg\max b_i$
  - charge winner i^* the 2nd highest report: $\arg\max_{j\neq i^*} b_j$
- **Claim:** It is in every i's best interest to report $b_i \equiv x_i$.

E.g. Computing the Max

- **Input:** $x_1, x_2, \ldots, x_n$
- **Goal**: compute $\max(x_1,\ldots,x_n)$
- **Algorithm:** Trivial
- But what if inputs are strategic?
  - suppose input i has value x_i for being selected&algorithm doesn't know x_i
  - facing trivial algorithm, every input reports $+\infty$
- **A better Algorithm [Vickrey'61 ]:**
  - collect reported inputs: $b_1,\ldots, b_n$ (can't enforce $b_i=x_i$ a priori)
  - select $i^* = \arg\max b_i$
  - charge winner i^* the 2nd highest report: $\arg\max_{j\neq i^*} b_j$
- **Claim:** It is in every i's best interest to report $b_i \equiv x_i$.

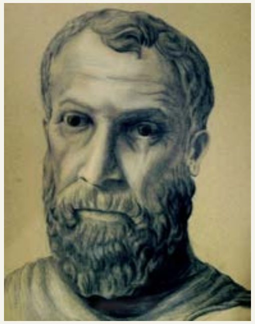
➔ **Vickrey auction** is the new **max**.

E.g. Computing the Max (cont'd)



Example: Solon's Law of *Antidosis* - 6th century BC

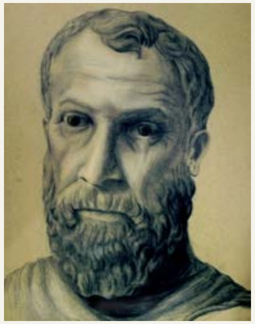
E.g. Computing the Max (cont'd)



Example: Solon's Law of *Antidosis* - 6th century BC

- Big Public Expenses: theater, gymnastics, ship building, etc
  - theater plays, festivals, gymnastics: ~ 3000 drachmas
  - ship building, and other military expenses: 4-6000 drachmas

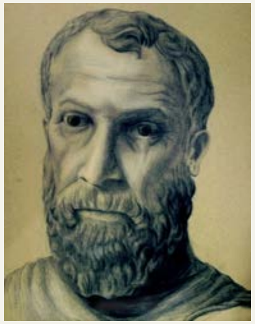
E.g. Computing the Max (cont'd)



Example: Solon's Law of *Antidosis* - 6th century BC

- Big Public Expenses: theater, gymnastics, ship building, etc
  - theater plays, festivals, gymnastics: ~ 3000 drachmas
  - ship building, and other military expenses: 4-6000 drachmas
- 1200 rich Athenians were eligible to pay
  - a committee selected a subset of them to cover these expenses.

E.g. Computing the Max (cont'd)

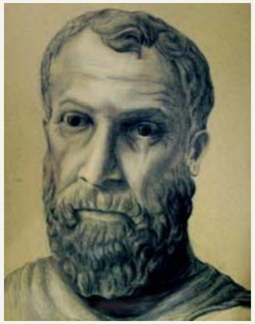


Example: Solon's Law of *Antidosis* - 6th century BC

- Big Public Expenses: theater, gymnastics, ship building, etc
  - theater plays, festivals, gymnastics: ~ 3000 drachmas
  - ship building, and other military expenses: 4-6000 drachmas
- 1200 rich Athenians were eligible to pay
  - a committee selected a subset of them to cover these expenses.

- A selected Athenian A who didn't want to pay for an assigned public expense, could exercise the law of **antidosis (property exchange)**, as follows:

E.g. Computing the Max (cont'd)

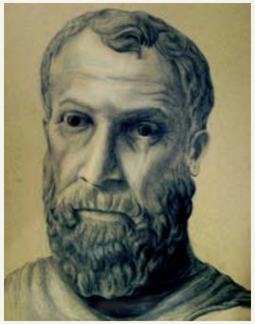


Example: Solon's Law of *Antidosis* - 6th century BC

- Big Public Expenses: theater, gymnastics, ship building, etc
  - theater plays, festivals, gymnastics: ~ 3000 drachmas
  - ship building, and other military expenses: 4-6000 drachmas
- 1200 rich Athenians were eligible to pay
  - a committee selected a subset of them to cover these expenses.

- A selected Athenian A who didn't want to pay for an assigned public expense, could exercise the law of **antidosis (property exchange)**, as follows:
  - **1st Step:** Athenian A names a non-selected Athenian B that he claims is richer, asking him to pay instead.

E.g. Computing the Max (cont'd)

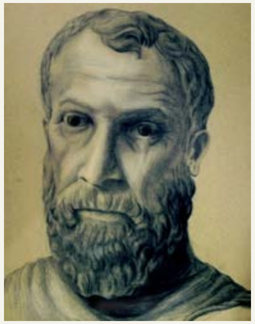


Example: Solon's Law of *Antidosis* - 6th century BC

- Big Public Expenses: theater, gymnastics, ship building, etc
  – theater plays, festivals, gymnastics: ~ 3000 drachmas
  – ship building, and other military expenses: 4-6000 drachmas
- 1200 rich Athenians were eligible to pay
  - a committee selected a subset of them to cover these expenses.

- A selected Athenian A who didn't want to pay for an assigned public expense, could exercise the law of **antidosis (property exchange)**, as follows:
  – **1st Step:** Athenian A names a non-selected Athenian B that he claims is richer, asking him to pay instead.
  – **2nd Step:** If B accepts to pay, then all good. Otherwise, A has the right to propose an exchange of properties with B before paying.

E.g. Computing the Max (cont'd)

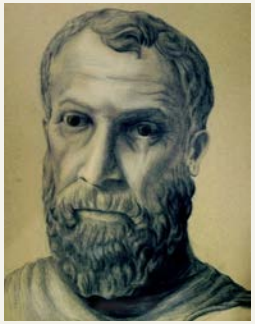


Example: Solon's Law of *Antidosis* - 6th century BC

- Big Public Expenses: theater, gymnastics, ship building, etc
  - theater plays, festivals, gymnastics: ~ 3000 drachmas
  - ship building, and other military expenses: 4-6000 drachmas
- 1200 rich Athenians were eligible to pay
  - a committee selected a subset of them to cover these expenses.

- A selected Athenian A who didn't want to pay for an assigned public expense, could exercise the law of **antidosis (property exchange)**, as follows:
  - **1st Step:** Athenian A names a non-selected Athenian B that he claims is richer, asking him to pay instead.
  - **2nd Step:** If B accepts to pay, then all good. Otherwise, A has the right to propose an exchange of properties with B before paying.
  - [If B doesn't accept the exchange, they go to court]

E.g. Computing the Max (cont'd)



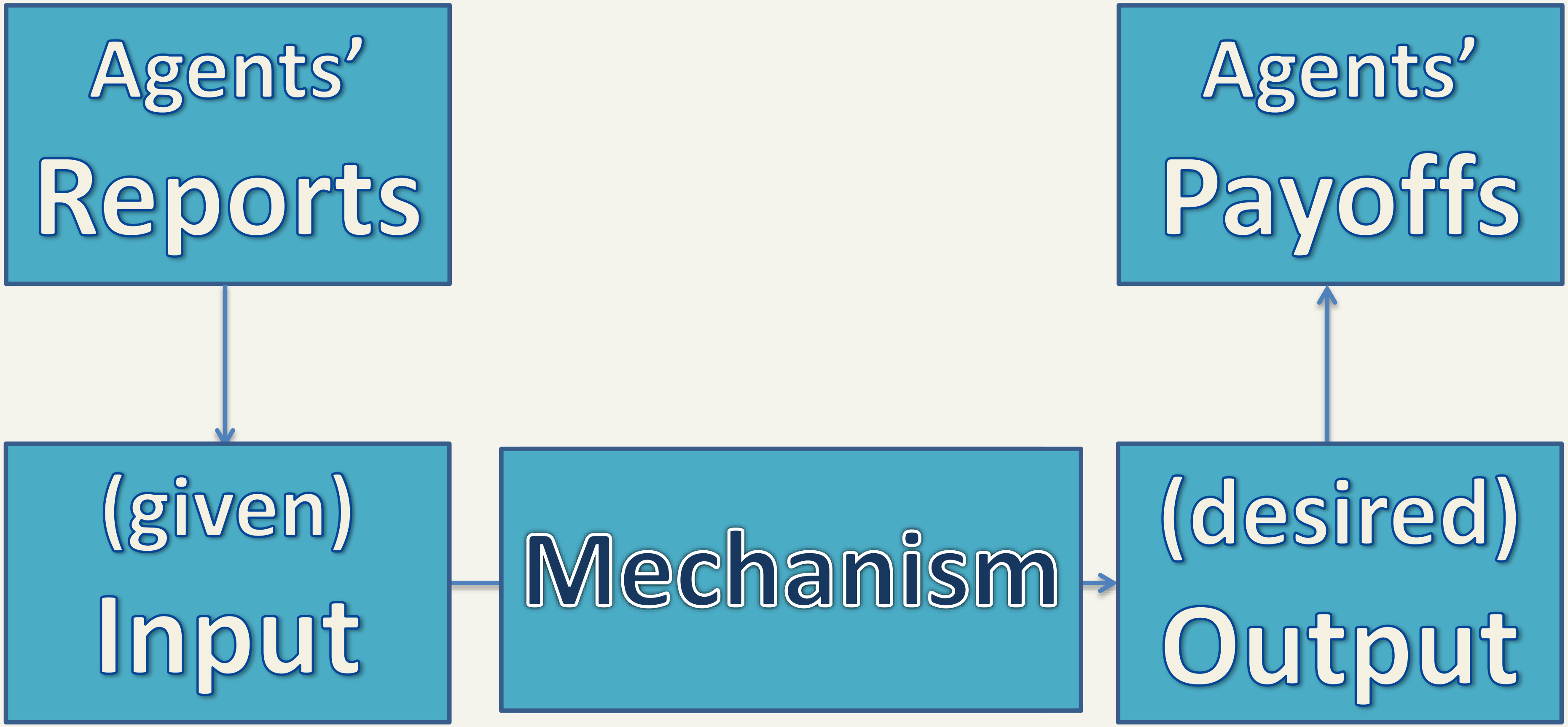
Example: Solon's Law of *Antidosis* - 6th century BC

- Big Public Expenses: theater, gymnastics, ship building, etc
  - theater plays, festivals, gymnastics: ~ 3000 drachmas
  - ship building, and other military expenses: 4-6000 drachmas
- 1200 rich Athenians were eligible to pay
  - a committee selected a subset of them to cover these expenses.

- A selected Athenian A who didn't want to pay for an assigned public expense, could exercise the law of **antidosis (property exchange)**, as follows:
  - **1st Step:** Athenian A names a non-selected Athenian B that he claims is richer, asking him to pay instead.
  - **2nd Step:** If B accepts to pay, then all good. Otherwise, A has the right to propose an exchange of properties with B before paying.
  - [If B doesn't accept the exchange, they go to court]

- **Outcome:** The richest subset of Athenians pays

Mechanism vs Algorithm Design



Mechanism vs Algorithm Design

[Nisan-Ronen'99]:

How much more difficult are optimization problems on "strategic" input compared to "honest" input?

Mechanism vs Algorithm Design

[Nisan-Ronen'99]:

How much more difficult are optimization problems on "strategic" input compared to "honest" input?

- **Information:**
  - what information does the mechanism have about the inputs?
  - what information do the inputs have about each other?
  - does the mechanism also have some private information whose release may influence the inputs' behavior (e.g. quality of a good in an auction)?

Mechanism vs Algorithm Design

[Nisan-Ronen'99]:

How much more difficult are optimization problems on "strategic" input compared to "honest" input?

- **Information:**
  - what information does the mechanism have about the inputs?
  - what information do the inputs have about each other?
  - does the mechanism also have some private information whose release may influence the inputs' behavior (e.g. quality of a good in an auction)?

- **Complexity:**
  - computational, communication, …
  - centralized: *complexity to run the mechanism*
    vs distributed*: complexity for each input to optimize own behavior*

The Menu

- Combinatorial Auctions
- Truthfulness vs Computation vs Communication
- Beyond the Truthfulness Barrier
- Meantime in a More Practical Universe..
- Algorithmic Mechanism Design for Learning Agents
- Discussion

The Menu

- **Combinatorial Auctions**
- Truthfulness vs Computation vs Communication
- Beyond the Truthfulness Barrier
- Meantime in a More Practical Universe..
- Algorithmic Mechanism Design for Learning Agents
- Discussion

Combinatorial Auctions

Combinatorial Auctions

Setting:

- Items: $[m]$
  - indivisible, heterogeneous, e.g. spectrum licenses
- Bidders: $[n]$
  - bidder i has (private) valuation $v_i: 2^{[m]} \to R_+$

Combinatorial Auctions

Setting:

- Items: $[m]$
  - indivisible, heterogeneous, e.g. spectrum licenses
- Bidders: $[n]$
  - bidder i has (private) valuation $v_i: 2^{[m]} \to R_+$

Goal:

- Choose: partition $[m] = S_1 \sqcup S_2 \sqcup \cdots \sqcup S_n$ and prices $(p_1, \ldots, p_n)$ so as to maximize the welfare from allocating set S_i to bidder i and charging him p_i, namely maximize $\sum_i v_i(S_i)$

Combinatorial Auctions

Setting:

- Items: $[\boldsymbol{m}]$
  - indivisible, heterogeneous, e.g. spectrum licenses
- Bidders: $[\boldsymbol{n}]$
  - bidder $\boldsymbol{i}$ has (private) valuation $\boldsymbol{v_i}: \boldsymbol{2^{[m]}} \rightarrow \boldsymbol{R_+}$

Goal:

- Choose: partition $[\boldsymbol{m}] = \boldsymbol{S_1} \sqcup \boldsymbol{S_2} \sqcup \cdots \sqcup \boldsymbol{S_n}$ and prices $(\boldsymbol{p_1}, \ldots, \boldsymbol{p_n})$ so as to maximize the welfare from allocating set $\boldsymbol{S_i}$ to bidder $\boldsymbol{i}$ and charging him $\boldsymbol{p_i}$, namely maximize $\sum_i \boldsymbol{v_i}(\boldsymbol{S_i})$

Issue: $\boldsymbol{v_i}$'s are unknown

Combinatorial Auctions

Setting:

- Items: $[m]$
  - indivisible, heterogeneous, e.g. spectrum licenses
- Bidders: $[n]$
  - bidder i has (private) valuation $v_i: 2^{[m]} \rightarrow R_+$

Goal:

- Choose: partition $[m] = S_1 \sqcup S_2 \sqcup \cdots \sqcup S_n$ and prices $(p_1, \ldots, p_n)$ so as to maximize the welfare from allocating set S_i to bidder i and charging him p_i, namely maximize $\sum_i v_i(S_i)$

Problem Solved!

- Vickrey-Clarke-Groves Mechanism

Combinatorial Auctions

Setting:

- Items: $[m]$
  - indivisible, heterogeneous, e.g. spectrum licenses
- Bidders: $[n]$
  - bidder i has (private) valuation $v_i: 2^{[m]} \to R_+$

Goal:

- Choose: partition $[m] = S_1 \sqcup S_2 \sqcup \cdots \sqcup S_n$ and prices $(p_1, \ldots, p_n)$ so as to maximize the welfare from allocating set S_i to bidder i and charging him p_i, namely maximize $\sum_i v_i(S_i)$

Problem Solved!

- Vickrey-Clarke-Groves Mechanism

1. Ask bidders to report valuation functions $\tilde{v}_1, \ldots \tilde{v}_n$

Combinatorial Auctions

Setting:

- Items: $[m]$
  - indivisible, heterogeneous, e.g. spectrum licenses
- Bidders: $[n]$
  - bidder i has (private) valuation $v_i: 2^{[m]} \to R_+$

Goal:

- Choose: partition $[m] = S_1 \sqcup S_2 \sqcup \cdots \sqcup S_n$ and prices $(p_1, \ldots, p_n)$ so as to maximize the welfare from allocating set S_i to bidder i and charging him p_i, namely maximize $\sum_i v_i(S_i)$

Problem Solved!

- Vickrey-Clarke-Groves Mechanism

1. Ask bidders to report valuation functions $\tilde{v}_1, \ldots \tilde{v}_n$
2. Choose $S_1, \ldots, S_n$ *maximizing* $\sum_i \tilde{v}_i(S_i)$

Combinatorial Auctions

Setting:

- Items: $[\boldsymbol{m}]$
  - indivisible, heterogeneous, e.g. spectrum licenses
- Bidders: $[\boldsymbol{n}]$
  - bidder $\boldsymbol{i}$ has (private) valuation $\boldsymbol{v_i}: \mathbf{2}^{[m]} \to \boldsymbol{R}_+$

Goal:

- Choose: partition $[\boldsymbol{m}] = \boldsymbol{S_1} \sqcup \boldsymbol{S_2} \sqcup \cdots \sqcup \boldsymbol{S_n}$ and prices $(\boldsymbol{p_1}, \ldots, \boldsymbol{p_n})$ so as to maximize the welfare from allocating set $\boldsymbol{S_i}$ to bidder $\boldsymbol{i}$ and charging him $\boldsymbol{p_i}$, namely maximize $\sum_i \boldsymbol{v_i}(\boldsymbol{S_i})$

Problem Solved!

- Vickrey-Clarke-Groves Mechanism

1. Ask bidders to report valuation functions $\tilde{\boldsymbol{v}}_1, \ldots \tilde{\boldsymbol{v}}_n$
2. Choose $\boldsymbol{S_1}, \ldots, \boldsymbol{S_n}$ *maximizing* $\sum_i \tilde{v}_i(\boldsymbol{S_i})$
3. Charge "Clarke payments"

Combinatorial Auctions

Setting:

- Items: $[m]$
  - indivisible, heterogeneous, e.g. spectrum licenses
- Bidders: $[n]$
  - bidder i has (private) valuation $v_i: 2^{[m]} \rightarrow R_+$

Goal:

- Choose: partition $[m] = S_1 \sqcup S_2 \sqcup \cdots \sqcup S_n$ and prices $(p_1, \ldots, p_n)$ so as to maximize the welfare from allocating set S_i to bidder i and charging him p_i, namely maximize $\sum_i v_i(S_i)$

Problem Solved!

- Vickrey-Clarke-Groves Mechanism

1. Ask bidders to report valuation functions $\tilde{v}_1, \ldots \tilde{v}_n$
2. Choose $S_1, \ldots, S_n$ *maximizing* $\sum_i \tilde{v}_i(S_i)$
3. Charge "Clarke payments"
   - ensures 1 is dominant strategy truthful, i.e. $\tilde{v}_i = v_i, \forall i$

Combinatorial Auctions

Setting:

- Items: $[m]$
  - indivisible, heterogeneous, e.g. spectrum licenses
- Bidders: $[n]$
  - bidder i has (private) valuation $v_i: 2^{[m]} \to R_+$

Goal:

- Choose: partition $[m] = S_1 \sqcup S_2 \sqcup \cdots \sqcup S_n$ and prices $(p_1, \dots, p_n)$ so as to maximize the welfare from allocating set S_i to bidder i and charging him p_i, namely maximize $\sum_i v_i(S_i)$

Problem Solved!

- Vickrey-Clarke-Groves Mechanism

1. Ask bidders to report valuation functions $\tilde{v}_1, \dots \tilde{v}_n$
2. Choose $S_1, \dots, S_n$ *maximizing* $\sum_i \tilde{v}_i(S_i)$
3. Charge "Clarke payments"
   - ensures 1 is dominant strategy truthful, i.e. $\tilde{v}_i = v_i, \forall i$
   - hence 2 chooses optimal allocation

VCG vs Communication vs Computation

[VCG '73]:

1. Ask bidders to report valuation functions $\tilde{v}_1, \ldots \tilde{v}_n$
2. Choose S *maximizing* $\sum_i \tilde{v}_i(S_i)$
3. Charge "Clarke payments"
   - ensures 1 is dominant strategy truthful, i.e. $\tilde{v}_i = v_i, \forall i$
   - hence 2 chooses optimal allocation

VCG vs Communication vs Computation

[VCG '73]:

1. Ask bidders to report valuation functions $\tilde{v}_1, \ldots \tilde{v}_n$
2. Choose S *maximizing* $\sum_i \tilde{v}_i(S_i)$
3. Charge "Clarke payments"
   - ensures 1 is dominant strategy truthful, i.e. $\tilde{v}_i = v_i, \forall i$
   - hence 2 chooses optimal allocation

Reality Check:

1. How are the valuation functions communicated in Step 1?
   - A valuation $v_i: 2^M \to R$ requires 2^m numbers to be specified

VCG vs Communication vs Computation

[VCG '73]:

1. Ask bidders to report valuation functions $\tilde{v}_1, \ldots \tilde{v}_n$
2. Choose S *maximizing* $\sum_i \tilde{v}_i(S_i)$
3. Charge "Clarke payments"
   - ensures 1 is dominant strategy truthful, i.e. $\tilde{v}_i = v_i, \forall i$
   - hence 2 chooses optimal allocation

Reality Check:

1. How are the valuation functions communicated in Step 1?
   - A valuation $v_i: 2^M \rightarrow R$ requires 2^m numbers to be specified
   - Solution? Consider only succinct v_i's or non-direct auctions interacting with bidders through value queries
     - *value* queries: "what is your value for bundle X?"

VCG vs Communication vs Computation

[VCG '73]:

1. Ask bidders to report valuation functions $\tilde{v}_1, \ldots \tilde{v}_n$
2. Choose S *maximizing* $\sum_i \tilde{v}_i(S_i)$
3. Charge "Clarke payments"
   - ensures 1 is dominant strategy truthful, i.e. $\tilde{v}_i = v_i, \forall i$
   - hence 2 chooses optimal allocation

Reality Check:

1. How are the valuation functions communicated in Step 1?
   - A valuation $v_i: 2^M \rightarrow R$ requires 2^m numbers to be specified
   - Solution? Consider only succinct v_i's or non-direct auctions interacting with bidders through value queries
     - *value* queries: "what is your value for bundle X?"

2. Even when valuations are succinct or can be queried, what if Step 2 can only be approximately solved?
   - Using approximation algorithms destroys truthfulness of VCG

VCG vs Communication vs Computation

[VCG '73]:

1. Ask bidders to report valuation functions $\tilde{v}_1, \ldots \tilde{v}_n$
2. Choose S *maximizing* $\sum_i \tilde{v}_i(S_i)$
3. Charge "Clarke payments"
   - ensures 1 is dominant strategy truthful, i.e. $\tilde{v}_i = v_i, \forall i$
   - hence 2 chooses optimal allocation

Reality Check:

1. How are the valuation functions communicated in Step 1?
   - A valuation $v_i: 2^M \to R$ requires 2^m numbers to be specified
   - Solution? Consider only succinct v_i's or non-direct auctions interacting with bidders through value queries
     - *value* queries: "what is your value for bundle X?"

2. Even when valuations are succinct or can be queried, what if Step 2 can only be approximately solved?
   - Using approximation algorithms destroys truthfulness of VCG
   - Are there truthful, approximately optimal, computationally efficient mechanisms?

VCG vs Communication vs Computation

VCG vs Communication vs Computation

- **Def:** $f: 2^M \to R$ is **submodular** iff
  $$\forall S \subseteq T, j \notin T: f(S \cup \{j\}) - f(S) \geq f(T \cup \{j\}) - f(T)$$
- **[Vondrak'08]:** Consider a combinatorial auction with submodular bidders. With value query access to ***true*** bidder valuations can achieve $\left(1 - \frac{1}{e}\right)$-fraction of optimal welfare in polynomial (in both $\boldsymbol{m}$ and $\boldsymbol{n}$) #queries/time.

VCG vs Communication vs Computation

- **Def:** $f: 2^M \to R$ is **submodular** iff
$$\forall S \subseteq T, j \notin T: f(S \cup \{j\}) - f(S) \geq f(T \cup \{j\}) - f(T)$$
- **[Vondrak'08]:** Consider a combinatorial auction with submodular bidders. With value query access to ***true*** bidder valuations can achieve $\left(1 - \frac{1}{e}\right)$-fraction of optimal welfare in polynomial (in both $\boldsymbol{m}$ and $\boldsymbol{n}$) #queries/time.
- **[Dughmi-Vondrak'11]:** If a truthful mechanism makes ***value queries*** to bidders and guarantees $\left(\frac{1}{\boldsymbol{m}^{0.01}}\right)$-fraction of optimal welfare, then it must make exponentially many queries.

VCG vs Communication vs Computation

- **Def:** $f: 2^M \to R$ is **submodular** iff
$$\forall S \subseteq T, j \notin T: f(S \cup \{j\}) - f(S) \geq f(T \cup \{j\}) - f(T)$$

- **[Vondrak'08]:** Consider a combinatorial auction with submodular bidders. With value query access to ***true*** bidder valuations can achieve $\left(1 - \frac{1}{e}\right)$-fraction of optimal welfare in polynomial (in both $\boldsymbol{m}$ and $\boldsymbol{n}$) #queries/time.

- **[Dughmi-Vondrak'11]:** If a truthful mechanism makes ***value queries*** to bidders and guarantees $\left(\frac{1}{m^{0.01}}\right)$-fraction of optimal welfare, then it must make exponentially many queries.

- **[Dobzinski-Vondrak'12]:** Even if each bidder's valuation can be succinctly described (w/ poly($\boldsymbol{m}$) info), no poly-time truthful mechanism can get better than $\left(\frac{1}{n^{0.01}}\right)$-fraction of optimal welfare, unless **NP** ⊆**P/poly**.

VCG vs Communication vs Computation

- **Def:** $f: 2^M \to R$ is **submodular** iff
  $$\forall S \subseteq T, j \notin T: f(S \cup \{j\}) - f(S) \geq f(T \cup \{j\}) - f(T)$$
- **[Vondrak'08]:** Consider a combinatorial auction with submodular bidders. With value query access to ***true*** bidder valuations can achieve $\left(1 - \frac{1}{e}\right)$-fraction of optimal welfare in polynomial (in both m and n) #queries/time.
- **[Dughmi-Vondrak'11]:** If a truthful mechanism makes ***value queries*** to bidders and guarantees $\left(\frac{1}{m^{0.01}}\right)$-fraction of optimal welfare, then it must make exponentially many queries.
- **[Dobzinski-Vondrak'12]:** Even if each bidder's valuation can be succinctly described (w/ poly(m) info), no poly-time truthful mechanism can get better than $\left(\frac{1}{n^{0.01}}\right)$-fraction of optimal welfare, unless **NP** ⊆**P/poly**.
- [**Papadimitriou, Schapira, Singer'08; Buchfuhrer et al'10, Dughmi-Vondrak'11, Dobzinski'11,Dobzinski-Vondrak'12, Daniely, Schapira, Shahaf'15**]:
  - "Truthfulness is at odds with communication and approximation"

The Menu

- **Combinatorial Auctions**
- **Truthfulness vs Computation vs Communication**
- Beyond the Truthfulness Barrier
- Meantime in a More Practical Universe..
- Algorithmic Mechanism Design for Learning Agents
- Discussion

The Menu

- **Combinatorial Auctions**
- **Truthfulness vs Computation vs Communication**
- **Beyond the Truthfulness Barrier**
- Meantime in a More Practical Universe..
- Algorithmic Mechanism Design for Learning Agents
- Discussion

Overcoming the Truthfulness Barrier

Overcoming the Truthfulness Barrier

- Combine any subset of:
  1. more powerful queries, e.g. *demand queries*
     - "given item prices $(p_1, \ldots, p_m)$ what is $\mathbf{arg\,max}\, v(S) - \sum_{i \in S} p_i$?"
  2. *Bayesian assumptions*
     - assume v_i's are drawn from distributions
     - compete against expected optimal welfare

Overcoming the Truthfulness Barrier

- Combine any subset of:
  1. more powerful queries, e.g. *demand queries*
     - "given item prices $(p_1, \dots, p_m)$ what is $\arg\max v(S) - \sum_{i \in S} p_i$?"
  2. *Bayesian assumptions*
     - assume v_i's are drawn from distributions
     - compete against expected optimal welfare

- **[...,Dobzinski'16]:** Poly-time, $O\left(\sqrt{\log m}\right)$-approximately optimal, truthful mechanism using demand queries, for XOS-bidders.
  - XOS valuations: *max of additive valuations* ⊃ Submodular valuations

Overcoming the Truthfulness Barrier

- Combine any subset of:
  1. more powerful queries, e.g. *demand queries*
     - "given item prices $(p_1, \dots, p_m)$ what is $\arg\max v(S) - \sum_{i \in S} p_i$?"
  2. *Bayesian assumptions*
     - assume v_i's are drawn from distributions
     - compete against expected optimal welfare

- **[…,Dobzinski'16]:** Poly-time, $O\left(\sqrt{\log m}\right)$-approximately optimal, truthful mechanism using demand queries, for XOS-bidders.
  - XOS valuations: *max of additive valuations* ⊃ Submodular valuations
- **[Feldman,Gravin,Lucier'15]:**

  Bayesian assumption + Demand Queries $\Rightarrow \frac{1}{2} \cdot$ OPT for XOS bidders

Overcoming the Truthfulness Barrier

- Combine any subset of:
  1. more powerful queries, e.g. *demand queries*
     - "given item prices $(p_1, \dots, p_m)$ what is $\arg\max v(S) - \sum_{i \in S} p_i$?"
  2. *Bayesian assumptions*
     - assume v_i's are drawn from distributions
     - compete against expected optimal welfare

- **[…,Dobzinski'16]:** Poly-time, $O\left(\sqrt{\log m}\right)$-approximately optimal, truthful mechanism using demand queries, for XOS-bidders.
  – XOS valuations: *max of additive valuations* ⊃ Submodular valuations
- **[Feldman,Gravin,Lucier'15]:**

  Bayesian assumption + Demand Queries $\Rightarrow \frac{1}{2} \cdot$ OPT for XOS bidders
- Just Bayes: **[Hartline-Lucier'10, Bei, Huang'11, Hartline-Malekian-Kleinberg'11]** provide black-box reductions from mechanism to algorithm design

Overcoming the Truthfulness Barrier

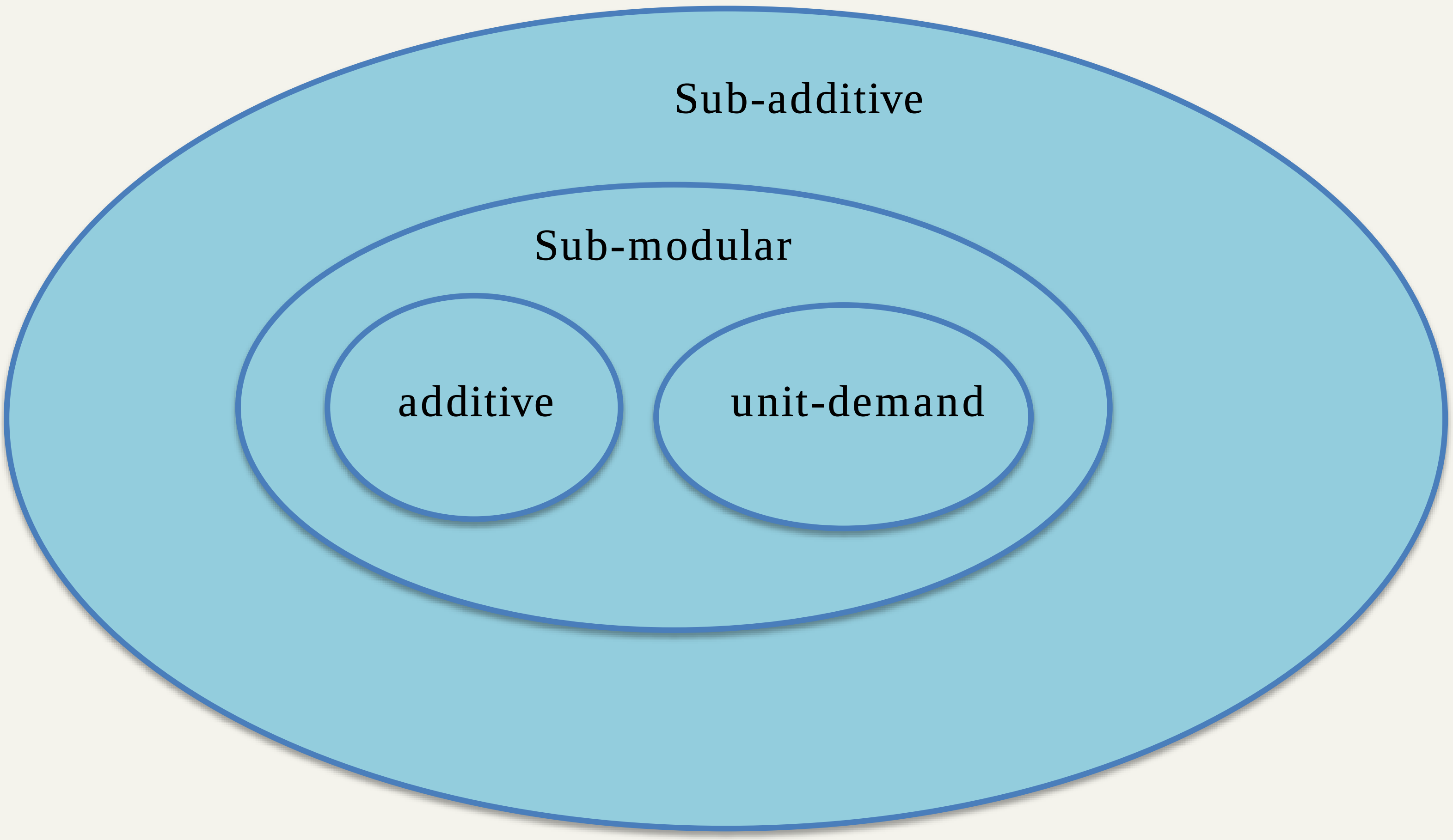
- Combine any subset of:
  1. more powerful queries, e.g. *demand queries*
     - "given item prices $(p_1, \ldots, p_m)$ what is $\arg\max v(S) - \sum_{i \in S} p_i$?"
  2. *Bayesian assumptions*
     - assume v_i's are drawn from distributions
     - compete against expected optimal welfare

- **[…,Dobzinski'16]:** Poly-time, $O\left(\sqrt{\log m}\right)$-approximately optimal, truthful mechanism using demand queries, for XOS-bidders.
  - XOS valuations: *max of additive valuations* ⊃ Submodular valuations
- **[Feldman,Gravin,Lucier'15]:**

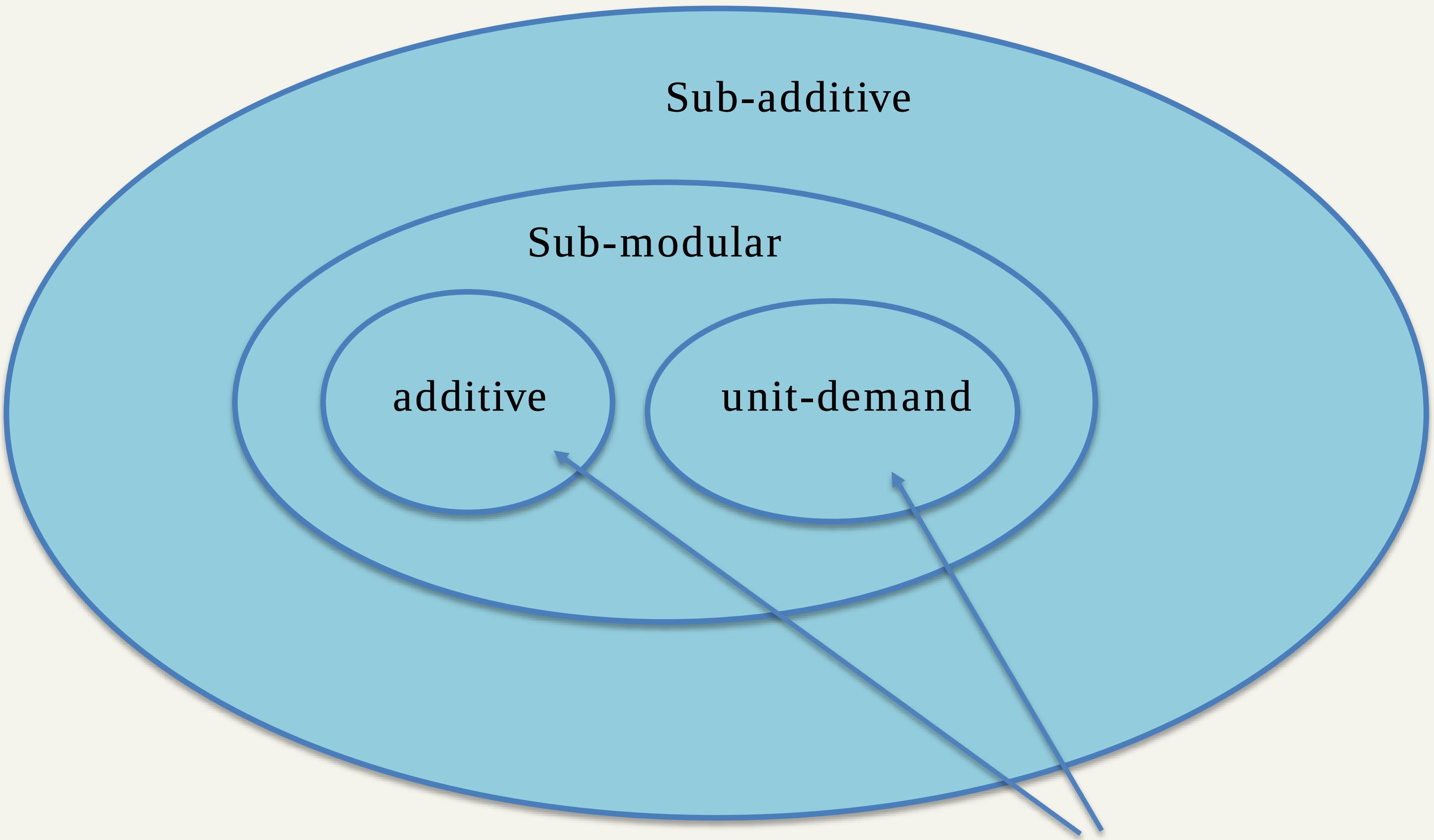
  Bayesian assumption + Demand Queries $\Rightarrow \frac{1}{2} \cdot \text{OPT}$ for XOS bidders

- Just Bayes: **[Hartline-Lucier'10, Bei, Huang'11, Hartline-Malekian-Kleinberg'11]** provide black-box reductions from mechanism to algorithm design
- **[Cai-Daskalakis-Weinberg'12-15]:** for any objective fn', e.g. revenue

Welfare Optimization (Summary)

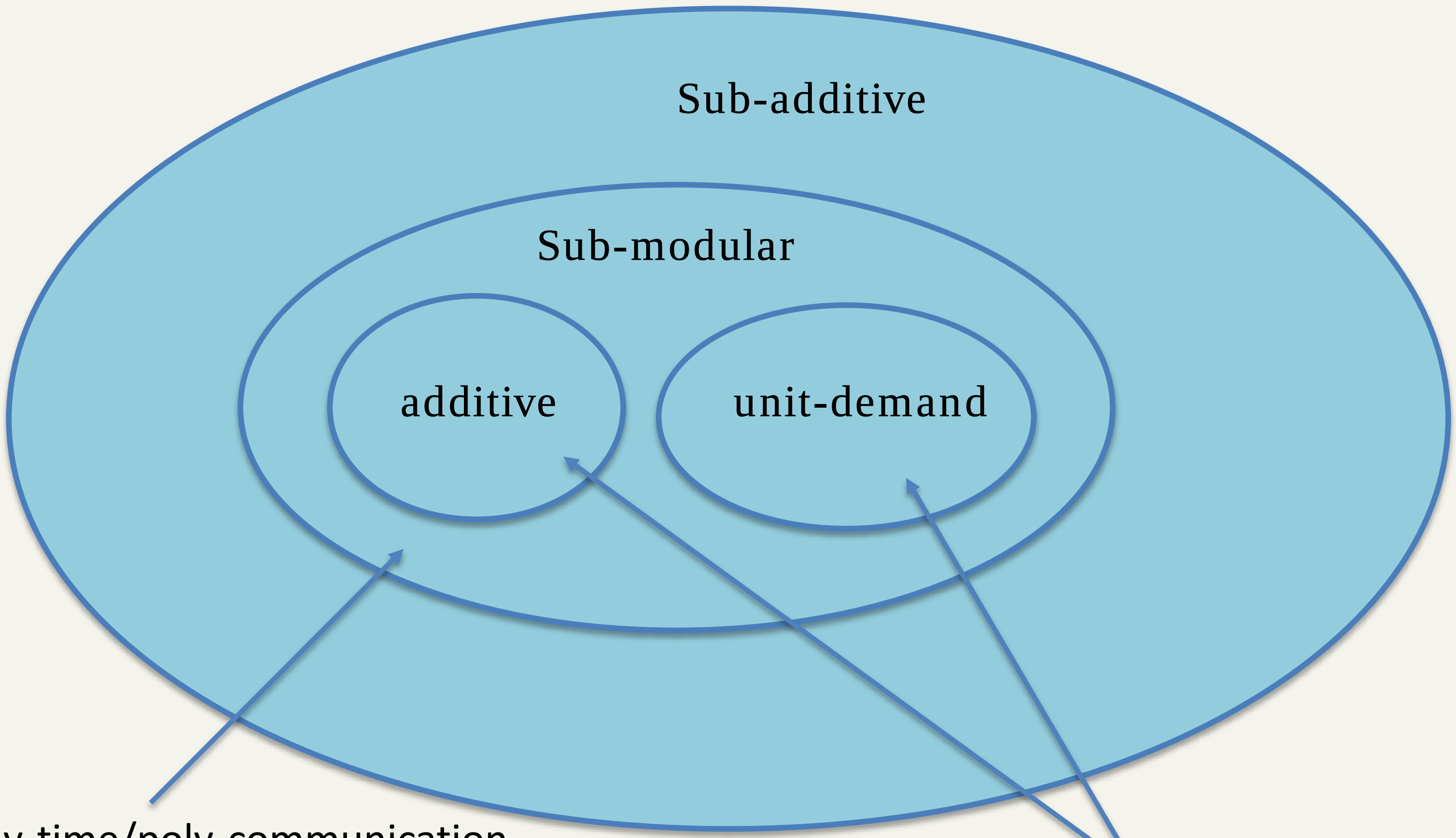


Welfare Optimization (Summary)



VCG gets OPT in poly-time, poly-communication and is truthful

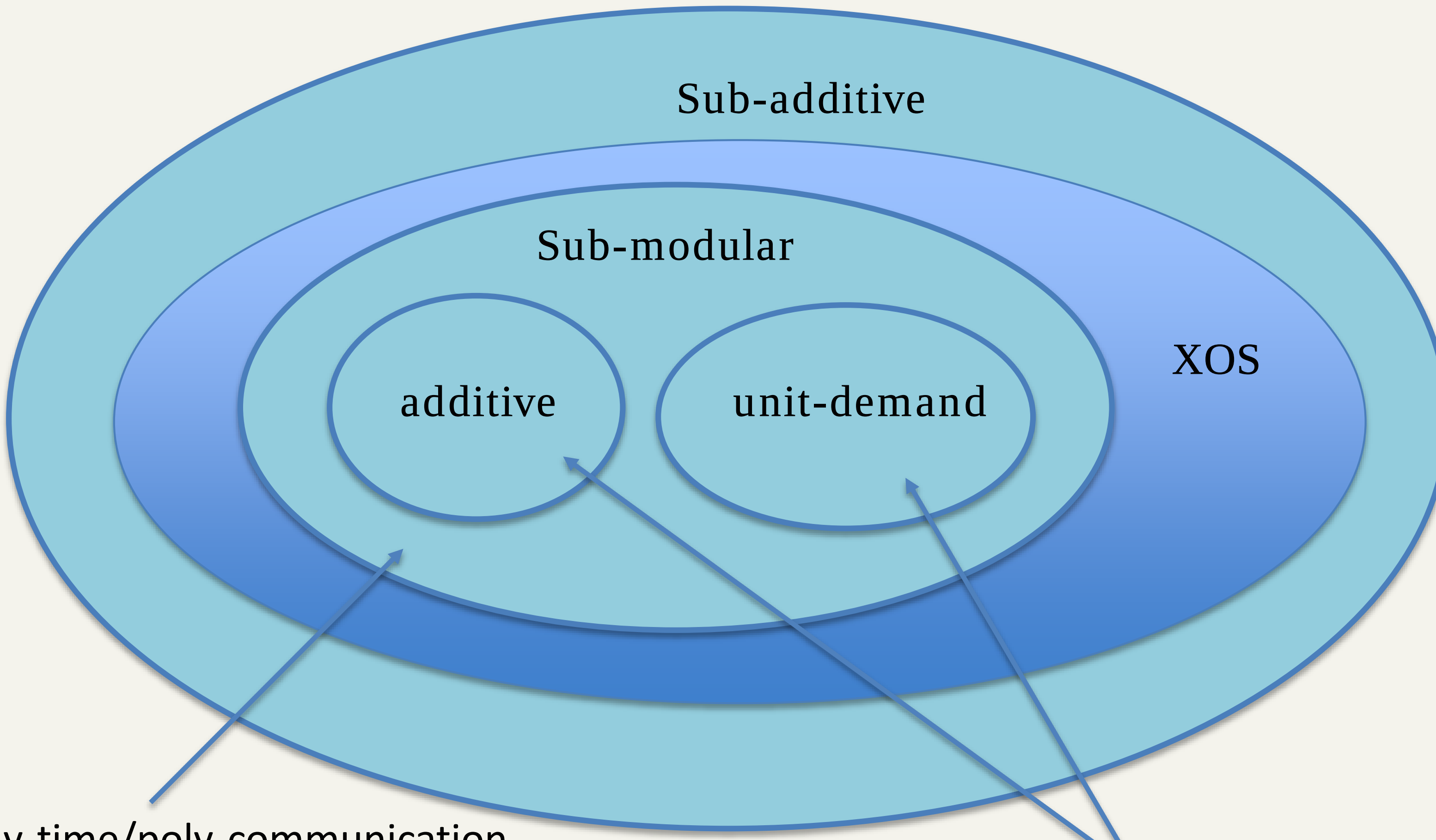
Welfare Optimization (Summary)



Truthful poly-time/poly-communication mechanisms lose polynomial factors

VCG gets OPT in poly-time, poly-communication and is truthful

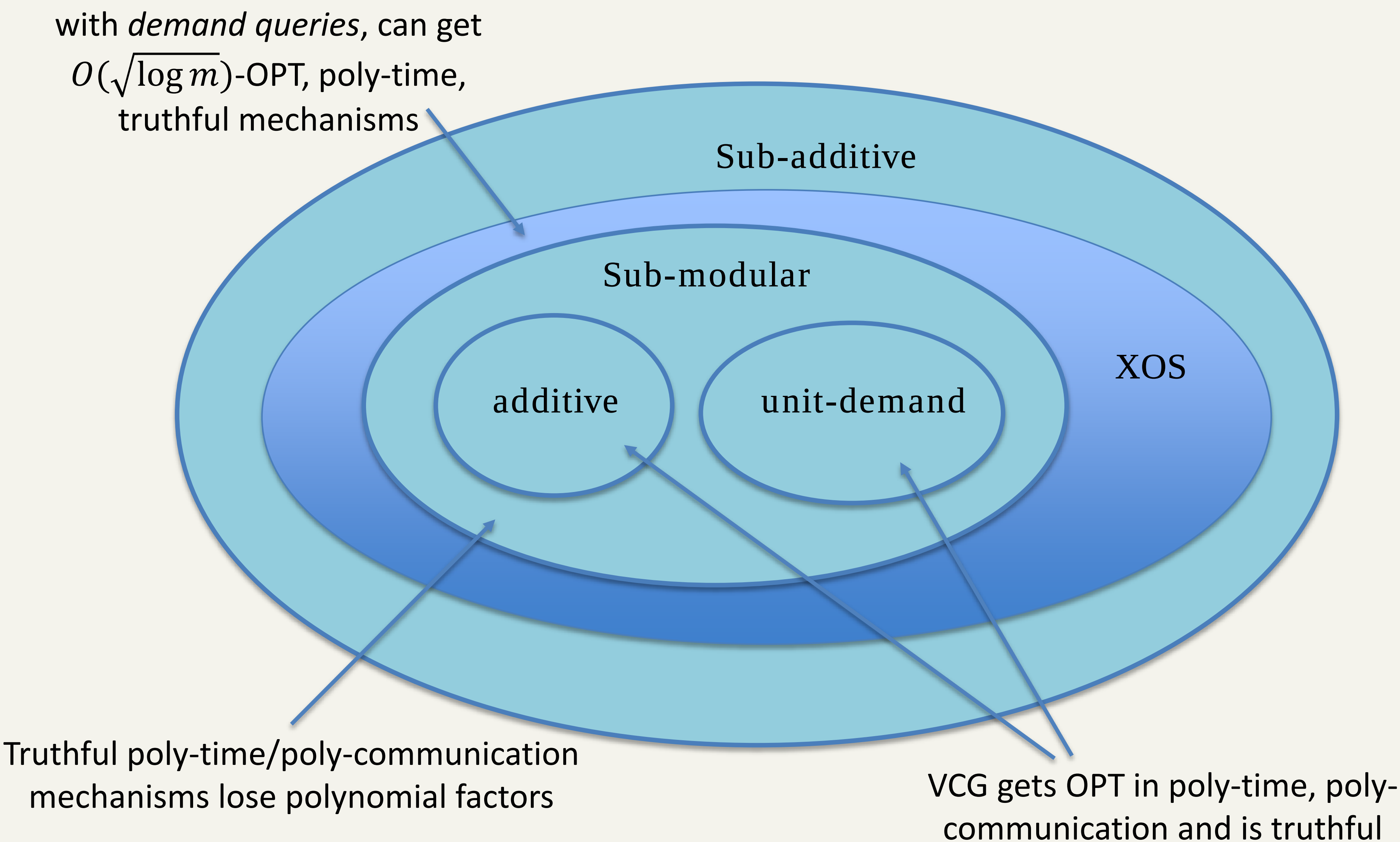
Welfare Optimization (Summary)



Truthful poly-time/poly-communication mechanisms lose polynomial factors

VCG gets OPT in poly-time, poly-communication and is truthful

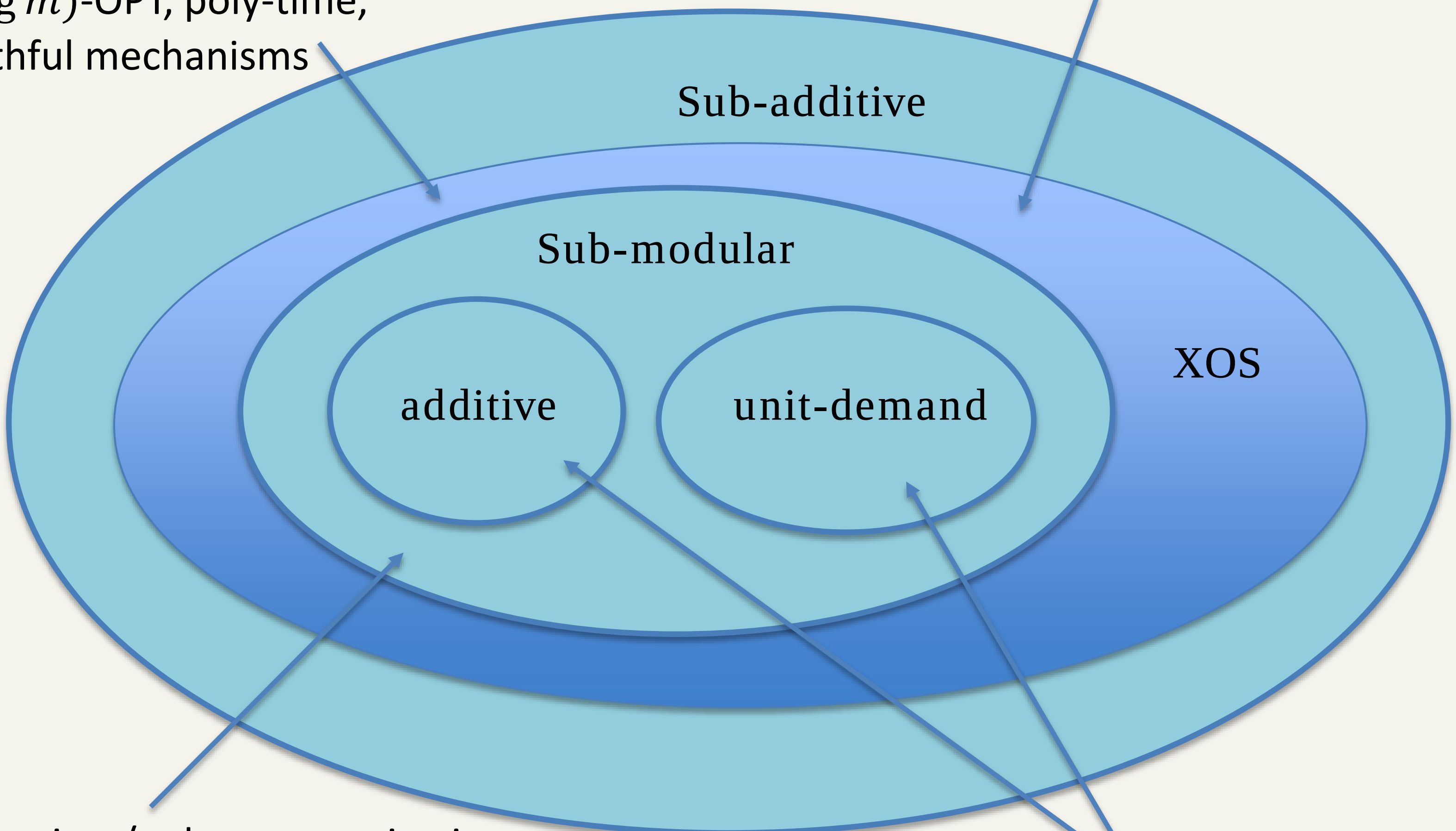
Welfare Optimization (Summary)



Welfare Optimization (Summary)

with *demand queries*, can get $O(\sqrt{\log m})$-OPT, poly-time, truthful mechanisms

demand queries + *Bayes:* 0.5-OPT, poly-time, truthful mechanisms



Truthful poly-time/poly-communication mechanisms lose polynomial factors

VCG gets OPT in poly-time, poly-communication and is truthful

The Menu

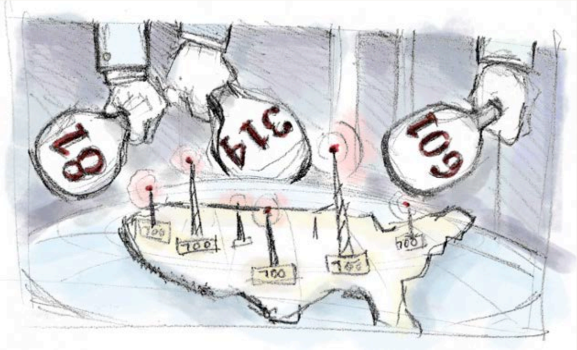
- **Combinatorial Auctions**
- **Truthfulness vs Computation vs Communication**
- **Beyond the Truthfulness Barrier**
- Meantime in a More Practical Universe..
- Algorithmic Mechanism Design for Learning Agents
- Discussion

The Menu

- **Combinatorial Auctions**
- **Truthfulness vs Computation vs Communication**
- **Beyond the Truthfulness Barrier**
- **Meantime in a More Practical Universe..**
- Algorithmic Mechanism Design for Learning Agents
- Discussion

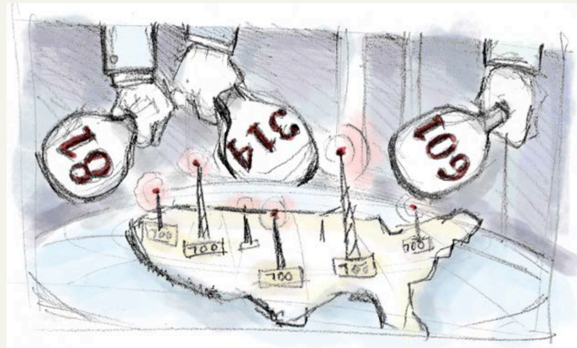
Meantime, in a more practical universe...

- ...auctions are used!



Meantime, in a more practical universe…

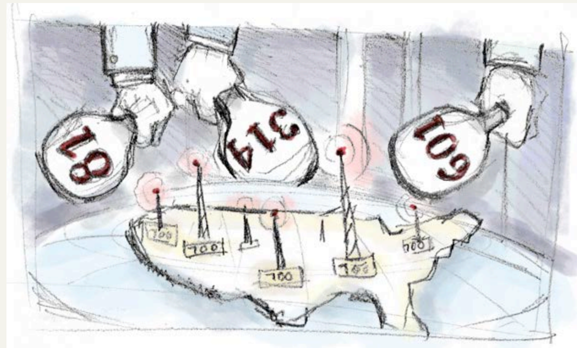
- …auctions are used!



- Non-Truthful Auction Environments
  - often simple auctions are composed sequentially and in parallel resulting in non-truthful overall mechanism

Meantime, in a more practical universe…

- …auctions are used!

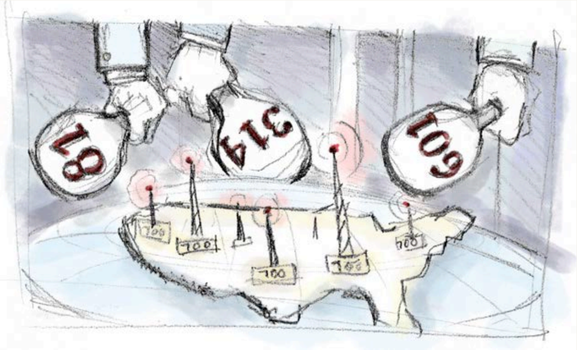


- Non-Truthful Auction Environments
  - often simple auctions are composed sequentially and in parallel resulting in non-truthful overall mechanism

- e.g. Simultaneous Second Price Auctions (SiSPAs)
  - sell m items in parallel using 2nd-price auctions
  - Mechanism is simple to describe, but non-truthful

Meantime, in a more practical universe…

- …auctions are used!

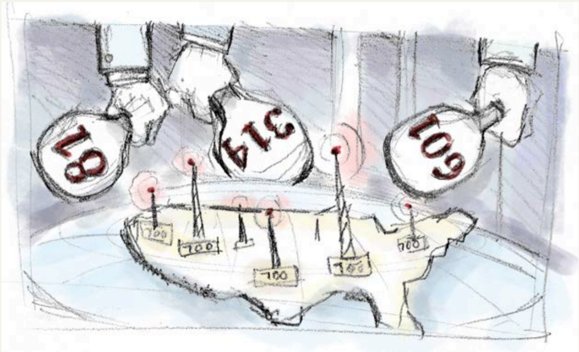


- Non-Truthful Auction Environments
  - often simple auctions are composed sequentially and in parallel resulting in non-truthful overall mechanism

- e.g. Simultaneous Second Price Auctions (SiSPAs)
  - sell m items in parallel using 2^{nd}-price auctions
  - Mechanism is simple to describe, but non-truthful
  - Challenging for non-additive bidders to bid as they need to anticipate how others will
    - going too strong, they may overpay; going too weak, they may lose items

Meantime, in a more practical universe…

- …auctions are used!



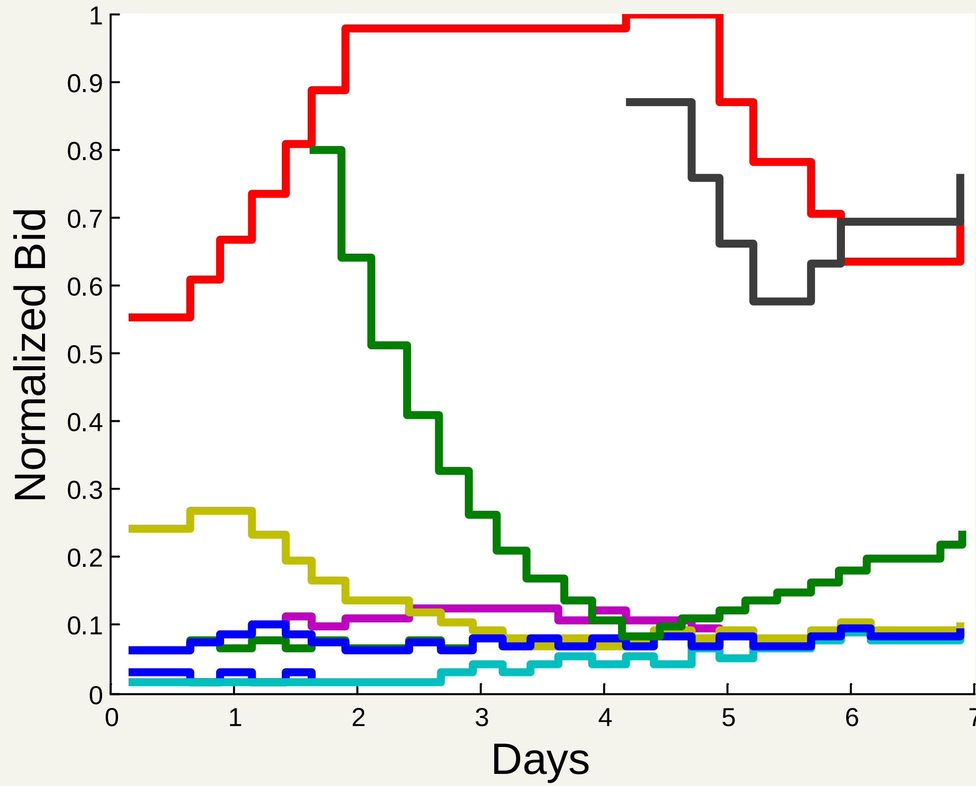
- Non-Truthful Auction Environments
  - often simple auctions are composed sequentially and in parallel resulting in non-truthful overall mechanism
- e.g. Simultaneous Second Price Auctions (SiSPAs)
  - sell m items in parallel using 2^{nd}-price auctions
  - Mechanism is simple to describe, but non-truthful
  - Challenging for non-additive bidders to bid as they need to anticipate how others will
    - going too strong, they may overpay; going too weak, they may lose items
- **Analytical challenge:** how do participants behave?

Bidder Behavior

- Classic microeconomic theory: Nash/Bayesian Nash equilibrium
  - requires heroic modeling assumptions
  - computationally hard and no decentralized dynamics converge **[Daskalakis, Goldberg, Papadimitriou'06, Hart, Mas-Colell '03]**

Bidder Behavior

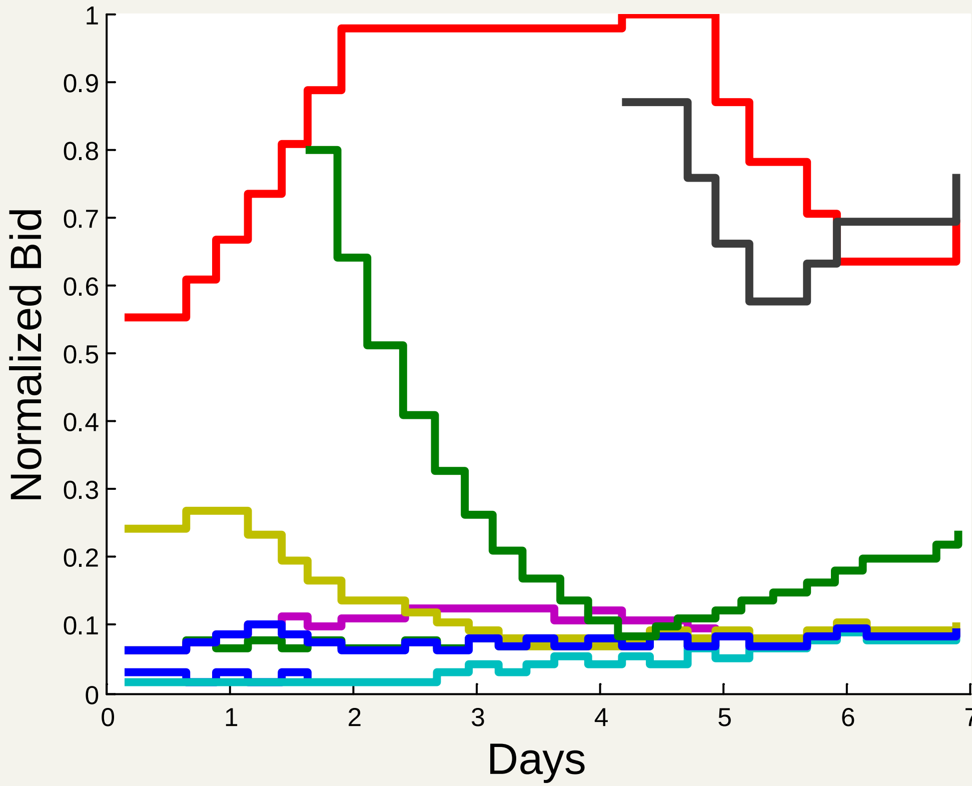
- Classic microeconomic theory: Nash/Bayesian Nash equilibrium
  - requires heroic modeling assumptions
  - computationally hard and no decentralized dynamics converge **[Daskalakis, Goldberg, Papadimitriou'06, Hart, Mas-Colell '03]**
- Stationarity of behavior inconsistent with data-sets



Data-Set from Microsoft's Bing
"*Econometrics for Learning Agents*" **[Nekipelov, Syrgkanis, Tardos'15]**

Bidder Behavior

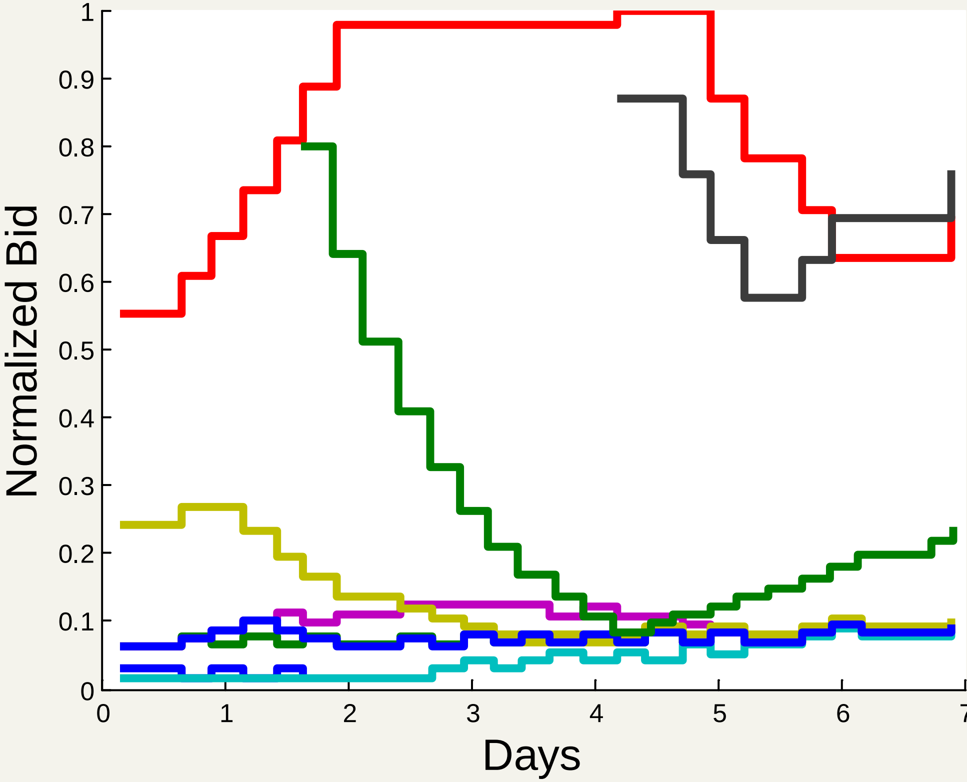
- Classic microeconomic theory: Nash/Bayesian Nash equilibrium
  - requires heroic modeling assumptions
  - computationally hard and no decentralized dynamics converge **[Daskalakis, Goldberg, Papadimitriou'06, Hart, Mas-Colell '03]**
- Stationarity of behavior inconsistent with data-sets
- **Natural approach:** out-of-equilibrium analysis
  - *mechanism design for learning agents*



Data-Set from Microsoft's Bing
"*Econometrics for Learning Agents*" **[Nekipelov, Syrgkanis, Tardos'15]**

Bidder Behavior

- Classic microeconomic theory: Nash/Bayesian Nash equilibrium
  - requires heroic modeling assumptions
  - computationally hard and no decentralized dynamics converge **[Daskalakis, Goldberg, Papadimitriou'06, Hart, Mas-Colell '03]**
- Stationarity of behavior inconsistent with data-sets
- **Natural approach:** out-of-equilibrium analysis
  - *mechanism design for learning agents*
  - fits well with certain auction settings such as online advertising



Data-Set from Microsoft's Bing
"*Econometrics for Learning Agents*" **[Nekipelov, Syrgkanis, Tardos'15]**

No-Regret Learning

- Fix mechanism M
  - suppose M is repeated T times with the same set of bidders

No-Regret Learning

- Fix mechanism M
  - suppose M is repeated T times with the same set of bidders

- **Def:** An algorithm choosing bid b_i^t for $t = 1, \dots, T$ is **"no-regret"** iff for any (adaptively chosen) sequence of $\left(b_{-i}^t\right)_{t=1,\dots,T}$:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[u_i\left(b_i^t, b_{-i}^t\right)\right] \geq \max_{b^*}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[u_i\left(b^*, b_{-i}^t\right)\right]\right) - o(1)$$

No-Regret Learning

- Fix mechanism M
  - suppose M is repeated T times with the same set of bidders

- **Def:** An algorithm choosing bid b_i^t for $t = 1, \dots, T$ is **"no-regret"** iff for any (adaptively chosen) sequence of $(b_{-i}^t)_{t=1,\dots,T}$:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[u_i(b_i^t, b_{-i}^t)] \geq \max_{b^*}\left(\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[u_i(b^*, b_{-i}^t)]\right) - o(1)$$

Expected avg utility under any behavior of other bidders

No-Regret Learning

- Fix mechanism M
  - suppose M is repeated T times with the same set of bidders

- **Def:** An algorithm choosing bid b_i^t for $t = 1, \dots, T$ is **"no-regret"** iff for any (adaptively chosen) sequence of $(b_{-i}^t)_{t=1,\dots,T}$:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[u_i(b_i^t, b_{-i}^t)] \geq \max_{b^*}\left(\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[u_i(b^*, b_{-i}^t)]\right) - o(1)$$

Expected avg utility under any behavior of other bidders

Best expected utility from a fixed bid

No-Regret Learning

- Fix mechanism M
  - suppose M is repeated T times with the same set of bidders

- **Def:** An algorithm choosing bid b_i^t for $t = 1, \dots, T$ is **"no-regret"** iff for any (adaptively chosen) sequence of $\left(b_{-i}^t\right)_{t=1,\dots,T}$:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[u_i\left(b_i^t, b_{-i}^t\right)\right] \geq \max_{b^*}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[u_i\left(b^*, b_{-i}^t\right)\right]\right) - o(1)$$

Expected avg utility under any behavior of other bidders

Best expected utility from a fixed bid

- **[Bik'99,CKS'08, BR'11, HKMN'11,FKL'12,ST'13, FFGL'13]:** In SiSPAs, the average welfare of the outcomes selected by a no-regret learning sequence is at least 0.25 OPT, even when bidders are **sub-additive**.*

No-Regret Learning

- Fix mechanism M
  - suppose M is repeated T times with the same set of bidders

- **Def:** An algorithm choosing bid b_i^t for $t = 1, \ldots, T$ is **"no-regret"** iff for any (adaptively chosen) sequence of $\left(b_{-i}^t\right)_{t=1,\ldots,T}$:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[u_i\left(b_i^t, b_{-i}^t\right)\right] \geq \max_{b^*}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[u_i\left(b^*, b_{-i}^t\right)\right]\right) - o(1)$$

Expected avg utility under any behavior of other bidders

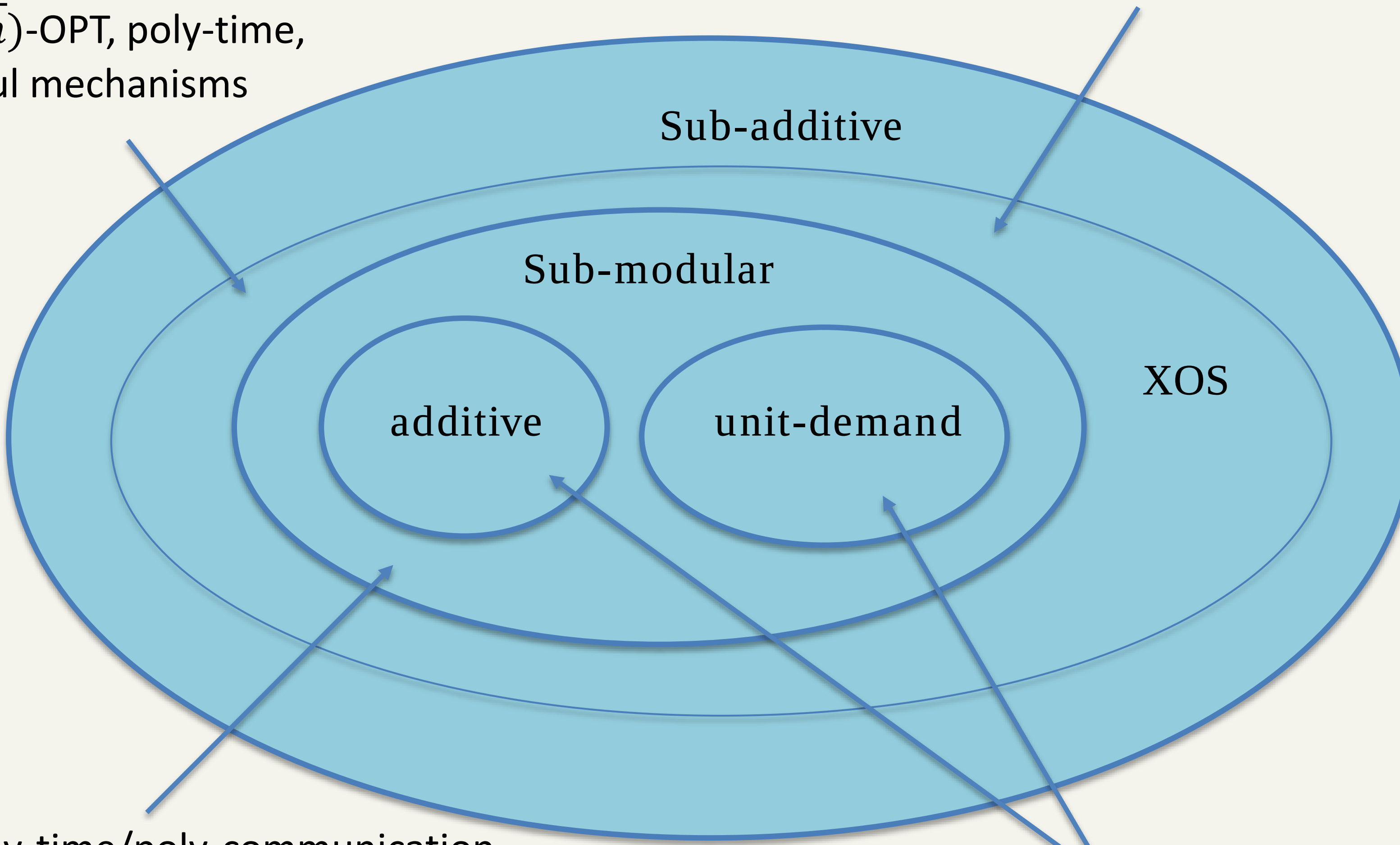
Best expected utility from a fixed bid

- **[Bik'99,CKS'08, BR'11, HKMN'11,FKL'12,ST'13, FFGL'13]:** In SiSPAs, the average welfare of the outcomes selected by a no-regret learning sequence is at least 0.25 OPT, even when bidders are **sub-additive**.*
  - constant factors hold for Simultaneous First Price auctions, and other types of smooth mechanisms
  - they also hold for full information Nash, incomplete info Bayes Nash equilibrium

Welfare Optimization (evolving summary)

with *demand queries*, can get $O(\sqrt{\log m})$-OPT, poly-time, truthful mechanisms

demand queries + *Bayes:* 0.5-OPT, poly-time, truthful mechanisms



Truthful poly-time/poly-communication mechanisms lose polynomial factors

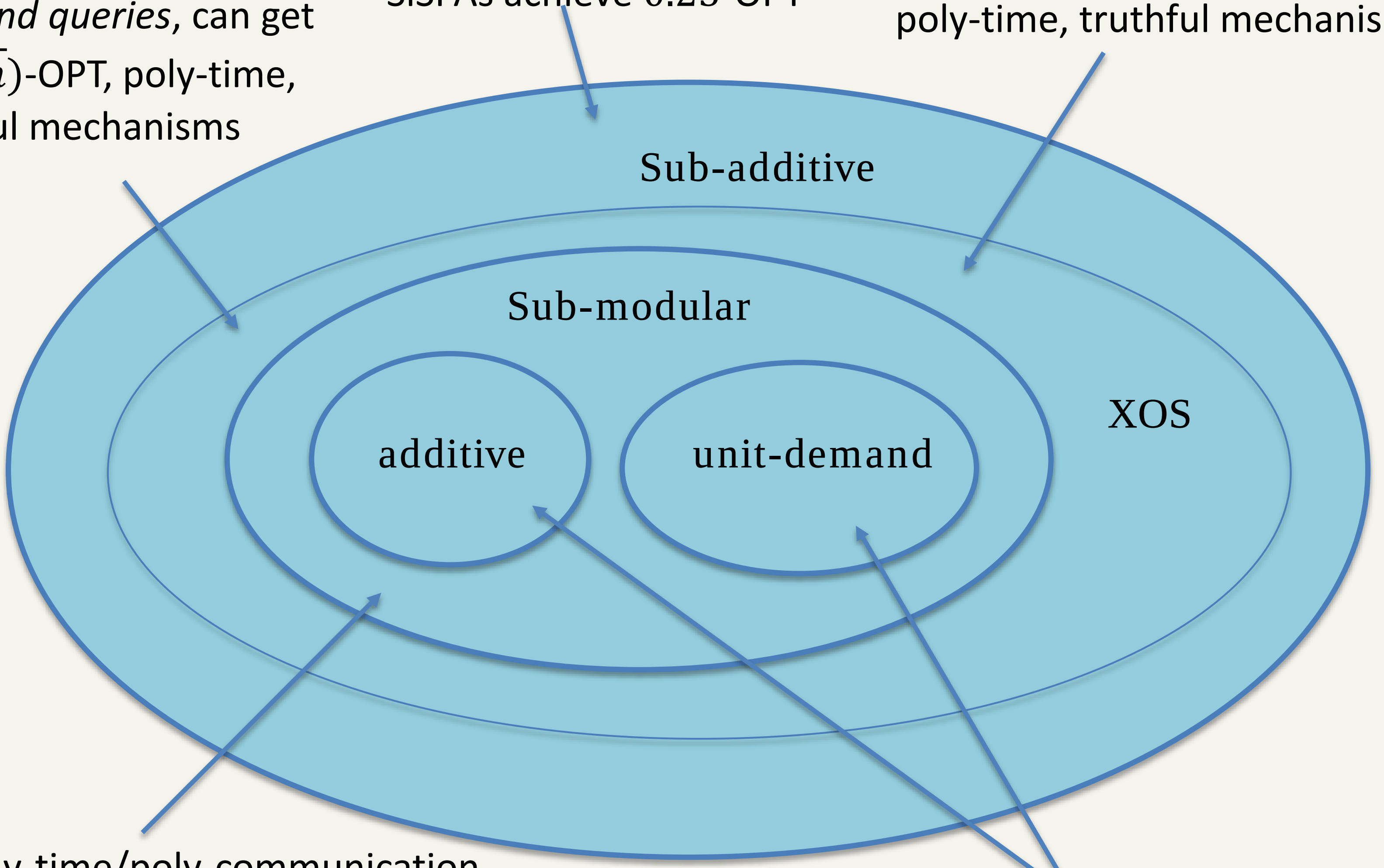
VCG gets OPT in poly-time, poly-communication and is truthful

Welfare Optimization (evolving summary)

under no-regret learning, SiSPAs achieve 0.25-OPT

demand queries + *Bayes:* 0.5-OPT, poly-time, truthful mechanisms

with *demand queries*, can get $O(\sqrt{\log m})$-OPT, poly-time, truthful mechanisms



Truthful poly-time/poly-communication mechanisms lose polynomial factors

VCG gets OPT in poly-time, poly-communication and is truthful

No-Regret Learning in Combinatorial Auctions

- **Important question:** Are equilibria or no-regret learning in SiSPAs, FiSPAS, or other non-truthful combinatorial auctions efficiently attainable?

No-Regret Learning in Combinatorial Auctions

- **Important question:** Are equilibria or no-regret learning in SiSPAs, FiSPAS, or other non-truthful combinatorial auctions efficiently attainable?

- **[Cai, Papadimitriou'14]:** In Bayesian setting answer is "no" even when distributions over valuations mix over additive and unit-demand valuations.

No-Regret Learning in Combinatorial Auctions

- **Important question:** Are equilibria or no-regret learning in SiSPAs, FiSPAS, or other non-truthful combinatorial auctions efficiently attainable?

- **[Cai, Papadimitriou'14]:** In Bayesian setting answer is "no" even when distributions over valuations mix over additive and unit-demand valuations.

- Is at least no-regret learning efficient?
  - **challenge:** #actions/experts is typically exponential in #items
  - e.g. in SiSPAs a no-regret learner has to compete well against all possible bid vectors

No-Regret Learning in Combinatorial Auctions

- **Important question:** Are equilibria or no-regret learning in SiSPAs, FiSPAS, or other non-truthful combinatorial auctions efficiently attainable?

- **[Cai, Papadimitriou'14]:** In Bayesian setting answer is "no" even when distributions over valuations mix over additive and unit-demand valuations.

- Is at least no-regret learning efficient?
  - **challenge:** #actions/experts is typically exponential in #items
  - e.g. in SiSPAs a no-regret learner has to compete well against all possible bid vectors

- **[Daskalakis-Syrganis'16]:** Unless $RP \supseteq NP$, there is no polynomial-time no-regret learning algorithm for bidding in SiSPAs, even for a unit-demand bidder.

Intractability of No-Regret Learning

- **[Daskalakis-Syrganis'16]:** Unless $RP \supseteq NP$, there is no polynomial-time no-regret learning algorithm for bidding in SiSPAs, even for a unit-demand bidder.
  - **Details:** Consider a unit-demand bidder with the same value v for all items, participating in T executions of a SiSPA with m items.
  - $\nexists$ no-regret learning algorithm that achieves regret poly$\left(\frac{1}{T}, v, m\right)$ and runs in time poly(T, v, m).

Intractability of No-Regret Learning

- **[Daskalakis-Syrganis'16]:** Unless $RP \supseteq NP$, there is no polynomial-time no-regret learning algorithm for bidding in SiSPAs, even for a unit-demand bidder.
    - **Details:** Consider a unit-demand bidder with the same value v for all items, participating in T executions of a SiSPA with m items.
    - $\nexists$ no-regret learning algorithm that achieves regret $\text{poly}\left(\frac{1}{T}, v, m\right)$ and runs in time $\text{poly}(T, v, m)$.
    - In particular, even allowing pseudo-polynomial dependence on v of both the runtime and the regret, it is still not possible to get the regret to drop like $\frac{poly(v,m)}{T^c}$ for any c>0, if learning algorithm runs in time $\text{poly}(T, m, v)$.

Intractability of No-Regret Learning

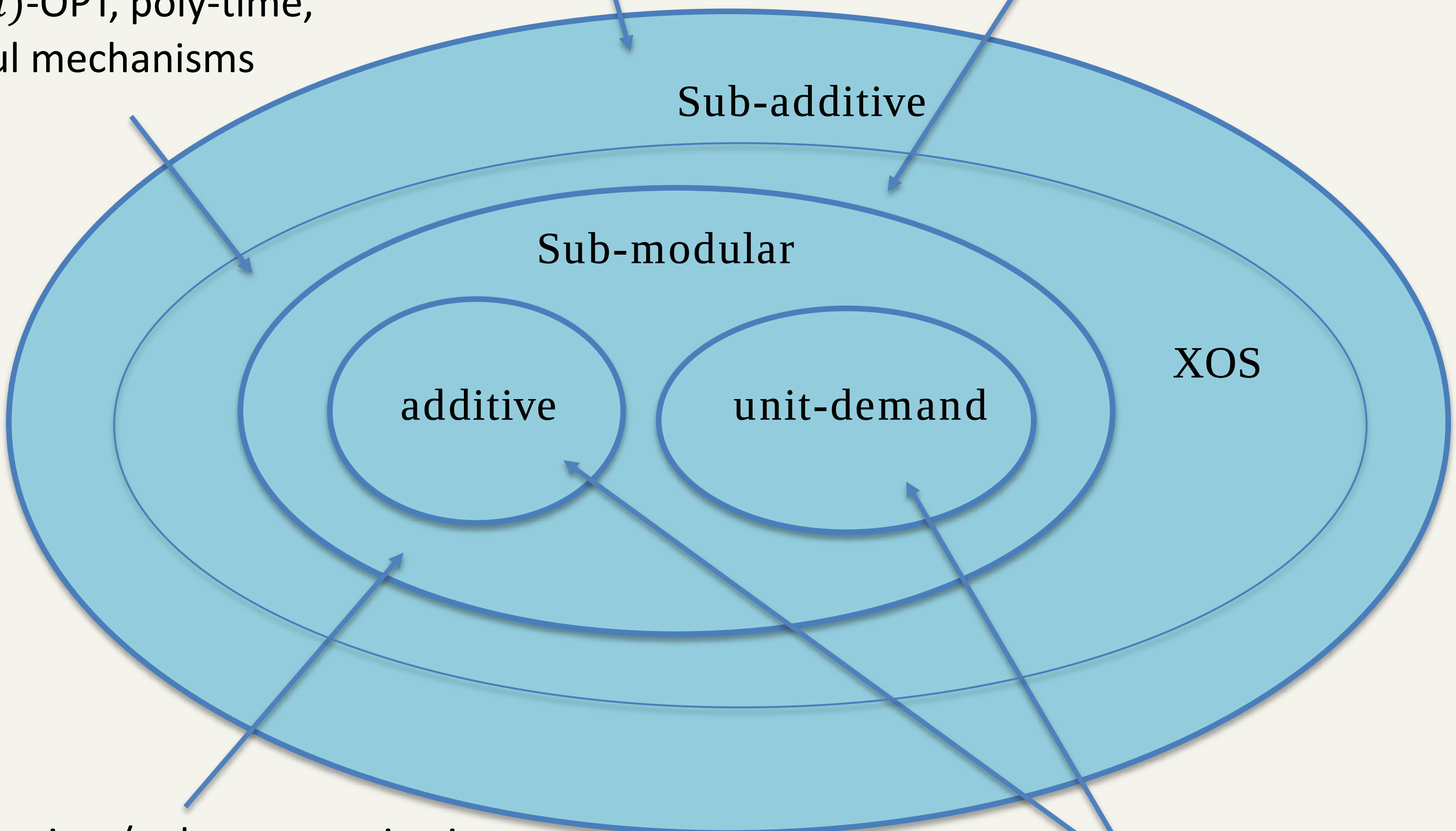
- **[Daskalakis-Syrganis'16]:** Unless $RP \supseteq NP$, there is no polynomial-time no-regret learning algorithm for bidding in SiSPAs, even for a unit-demand bidder.
  - **Details:** Consider a unit-demand bidder with the same value v for all items, participating in T executions of a SiSPA with m items.
  - $\nexists$ no-regret learning algorithm that achieves regret poly$\left(\frac{1}{T}, v, m\right)$ and runs in time poly(T, v, m).
  - In particular, even allowing pseudo-polynomial dependence on v of both the runtime and the regret, it is still not possible to get the regret to drop like $\frac{poly(v,m)}{T^c}$ for any c>0, if learning algorithm runs in time poly(T, m, v).
  - This is true even if our bidder plays against one stationary opponent, whose bids in every round are i.i.d. samples from an explicitly given distribution of bid vectors.

Welfare Optimization (evolving summary)

with *demand queries*, can get $O(\sqrt{\log m})$-OPT, poly-time, truthful mechanisms

under no-regret learning, SiSPAs achieve 0.25-OPT

demand queries + *Bayes:* 0.5-OPT, poly-time, truthful mechanisms



Truthful poly-time/poly-communication mechanisms lose polynomial factors

VCG gets OPT in poly-time, poly-communication and is truthful

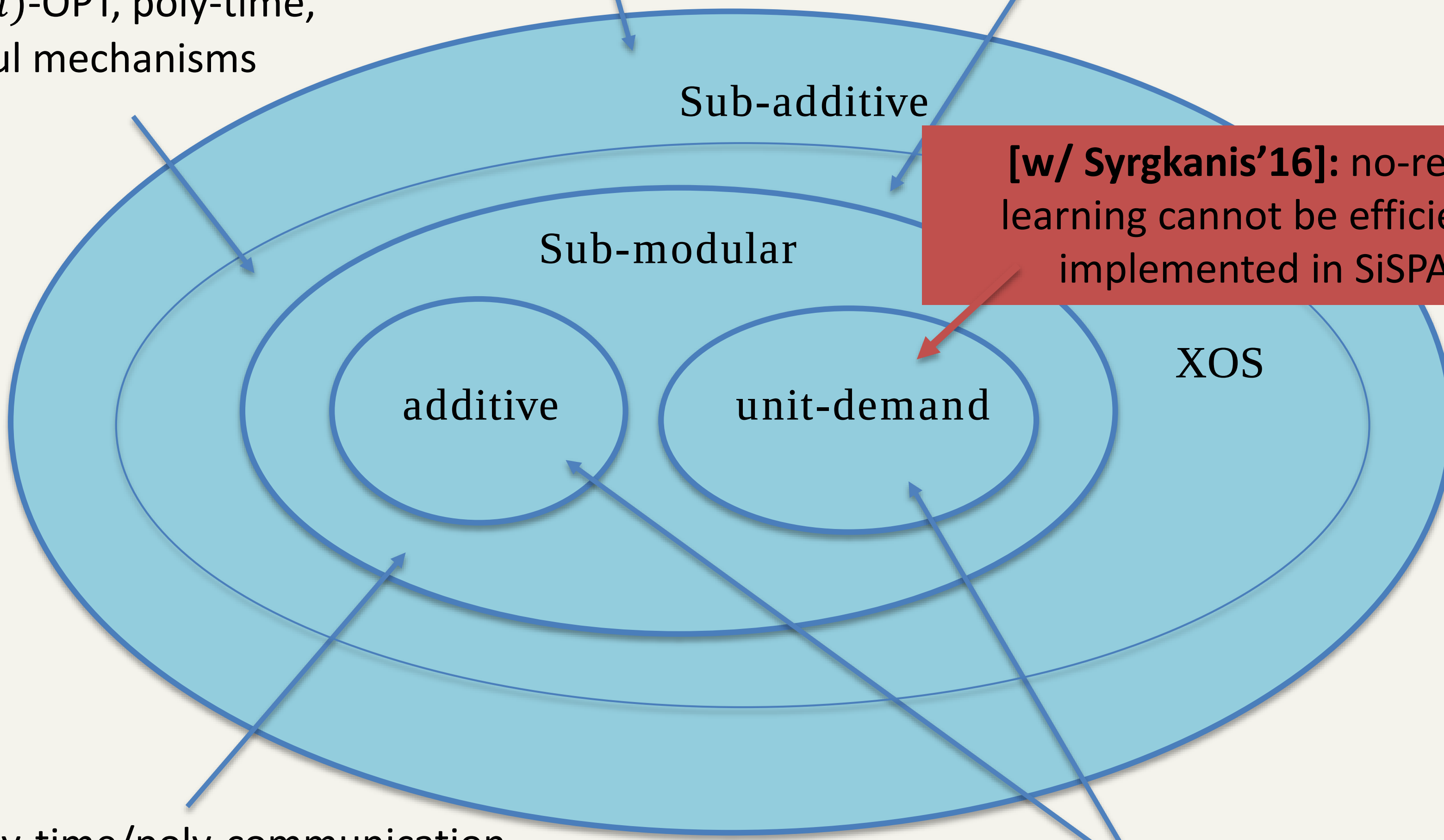
Welfare Optimization (evolving summary)

with *demand queries*, can get $O(\sqrt{\log m})$-OPT, poly-time, truthful mechanisms

under no-regret learning, SiSPAs achieve 0.25-OPT

demand queries + *Bayes:* 0.5-OPT, poly-time, truthful mechanisms

[w/ Syrgkanis'16]: no-regret learning cannot be efficiently implemented in SiSPAs



Truthful poly-time/poly-communication mechanisms lose polynomial factors

VCG gets OPT in poly-time, poly-communication and is truthful

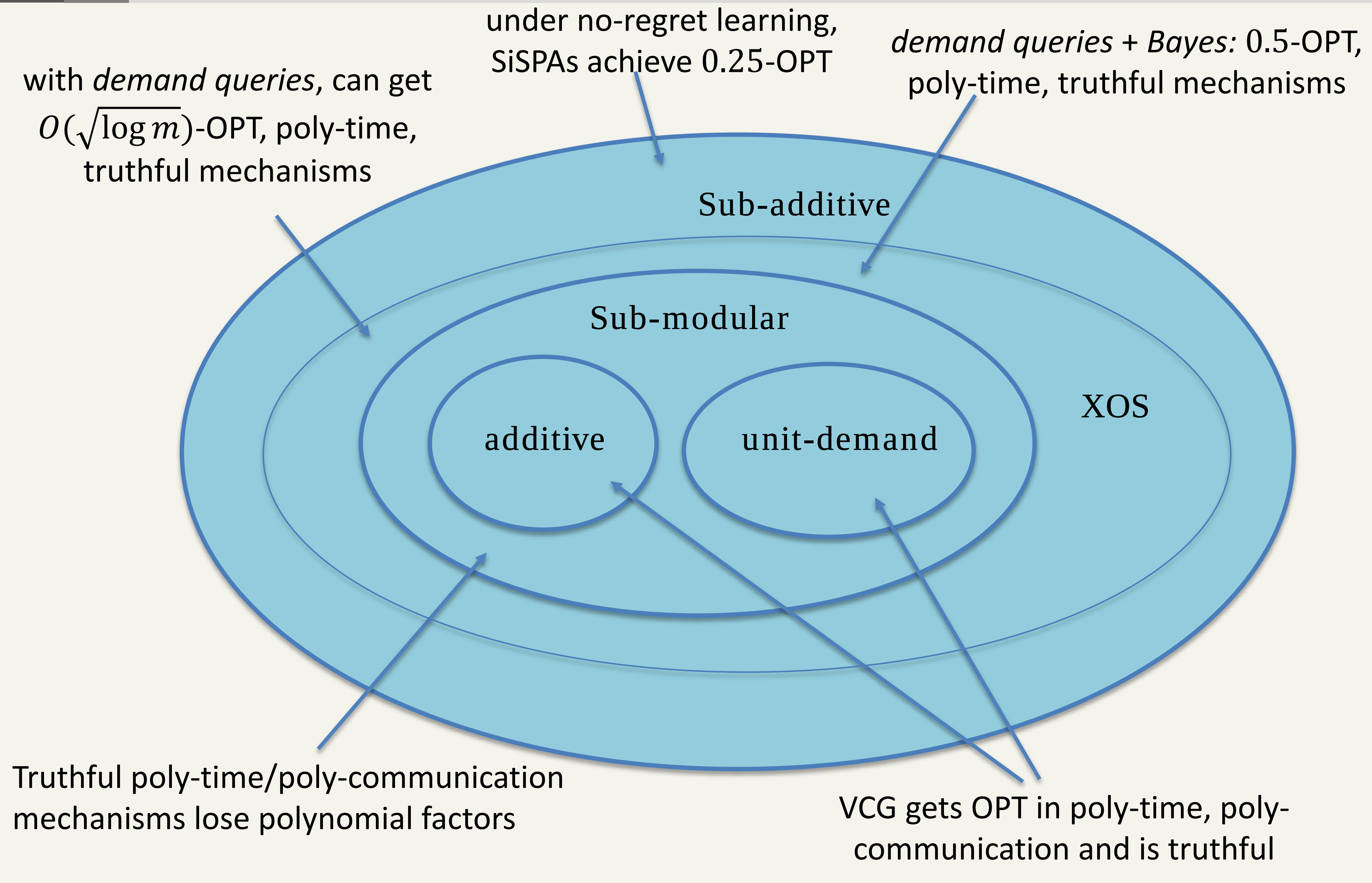
The Menu

- **Combinatorial Auctions**
- **Truthfulness vs Computation vs Communication**
- **Beyond the Truthfulness Barrier**
- **Meantime in a More Practical Universe..**
- Algorithmic Mechanism Design for Learning Agents
- Discussion

The Menu

- **Combinatorial Auctions**
- **Truthfulness vs Computation vs Communication**
- **Beyond the Truthfulness Barrier**
- **Meantime in a More Practical Universe..**
- **Algorithmic Mechanism Design for Learning Agents**
- Discussion

Welfare Optimization (evolving summary)

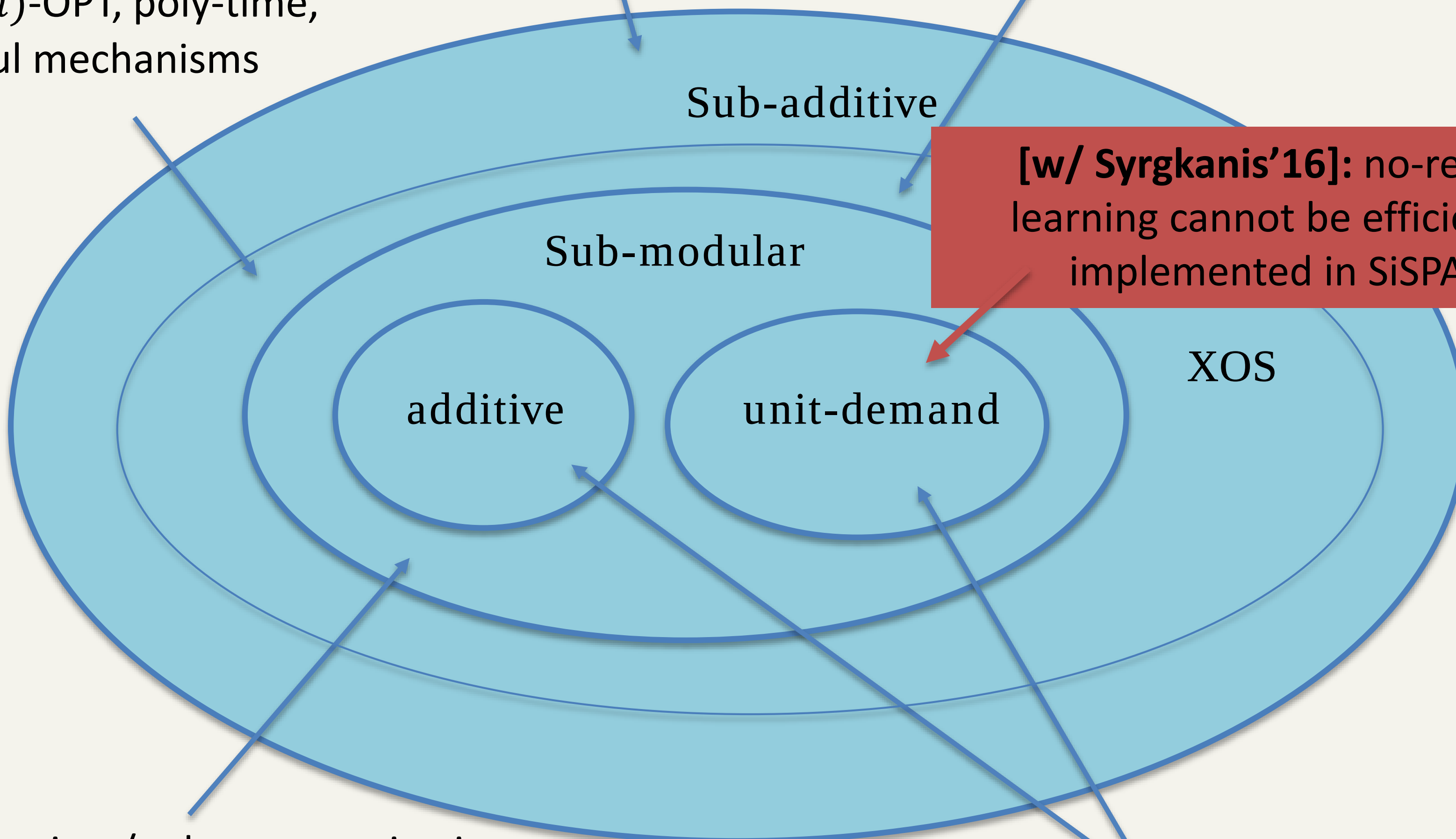


Welfare Optimization (evolving summary)

under no-regret learning, SiSPAs achieve 0.25-OPT

with *demand queries*, can get $O(\sqrt{\log m})$-OPT, poly-time, truthful mechanisms

demand queries + *Bayes:* 0.5-OPT, poly-time, truthful mechanisms

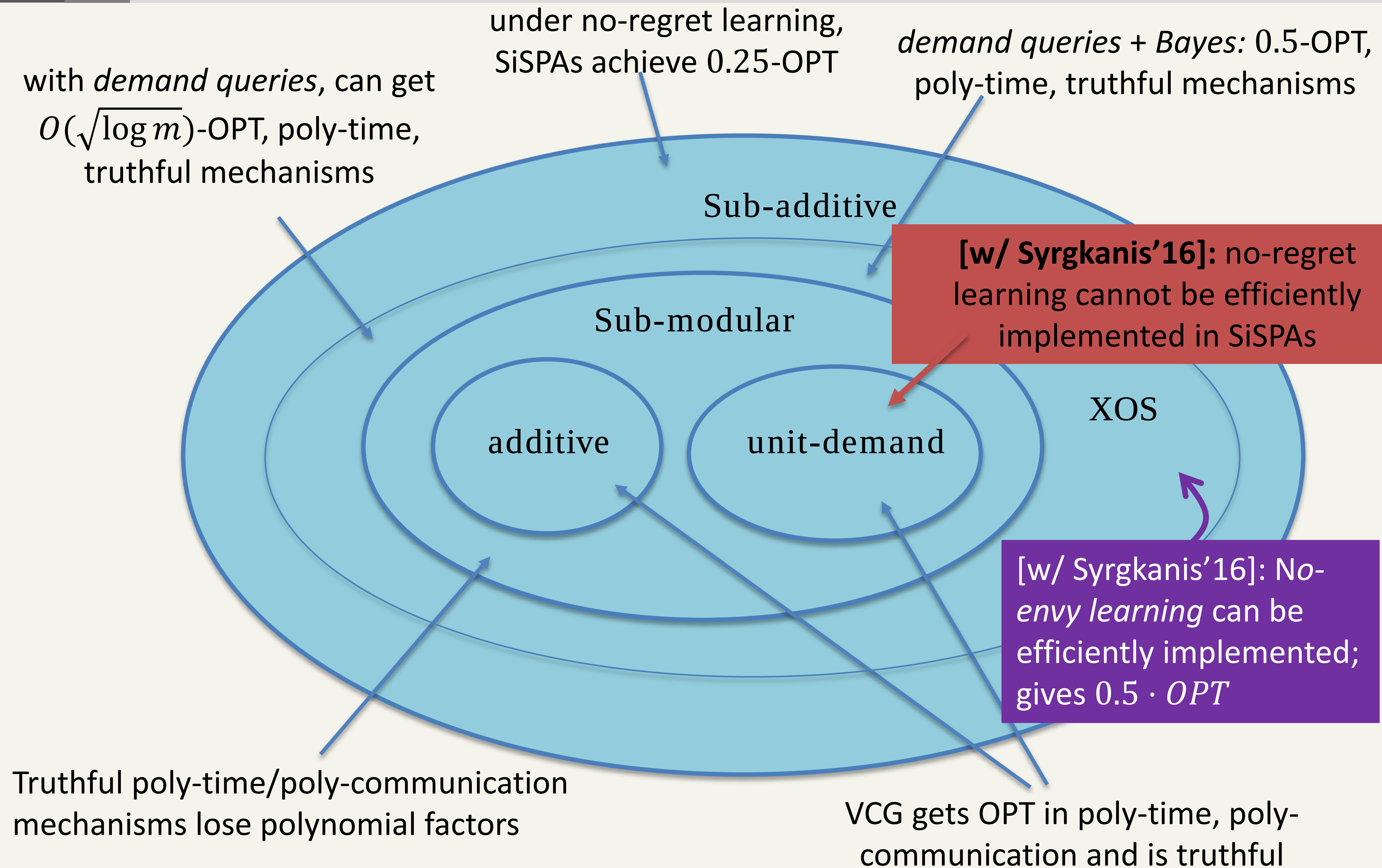


[w/ Syrgkanis'16]: no-regret learning cannot be efficiently implemented in SiSPAs

Truthful poly-time/poly-communication mechanisms lose polynomial factors

VCG gets OPT in poly-time, poly-communication and is truthful

Welfare Optimization (evolving summary)



No-Envy vs No-Regret Learning

- Fix mechanism M
  - Suppose n bidders engage in a repeated execution of mechanism M
  - b_i^t: bidder i's action in round t
    - in SiSPAs, this is a vector of bids on each item
    - in more complex mechanisms, more complex

No-Envy vs No-Regret Learning

- Fix mechanism M
  - Suppose n bidders engage in a repeated execution of mechanism M
  - b_i^t: bidder i's action in round t
    - in SiSPAs, this is a vector of bids on each item
    - in more complex mechanisms, more complex
- **Def:** An algorithm that chooses bid b_i^t for all $t = 1, \ldots, T$ is **"no-regret"** iff for any (adaptively chosen) $\left(b_{-i}^t\right)_{t=1,\ldots,T}$

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[u_i(b_i^t, b_{-i}^t)] \geq \max_{b^*}\left(\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[u_i(b^*, b_{-i}^t)]\right) - o(1)$$

No-Envy vs No-Regret Learning

- Fix mechanism M
  - Suppose n bidders engage in a repeated execution of mechanism M
  - b_i^t: bidder i's action in round t
    - in SiSPAs, this is a vector of bids on each item
    - in more complex mechanisms, more complex
- **Def:** An algorithm that chooses bid b_i^t for all $t = 1, \dots, T$ is **"no-regret"** iff for any (adaptively chosen) $\left(b_{-i}^t\right)_{t=1,\dots,T}$

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[u_i\left(b_i^t, b_{-i}^t\right)\right] \geq \max_{b^*}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[u_i\left(b^*, b_{-i}^t\right)\right]\right) - o(1)$$

- **Def:** An algorithm that chooses bid b_i^t for all $t = 1, \dots, T$ is **"no-envy"** iff for any (adaptively chosen) $\left(b_{-i}^t\right)_{t=1,\dots,T}$

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[u_i(b_i^t, b_{-i}^t)\right] \geq \max_{S^*}\left(v_i(S^*) - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[p_{S^*,i}\left(b_{-i}^t\right)\right]\right) - o(1)$$

No-Envy vs No-Regret Learning

- Fix mechanism M
  - Suppose n bidders engage in a repeated execution of mechanism M
  - b_i^t: bidder i's action in round t
    - in SiSPAs, this is a vector of bids on each item
    - in more complex mechanisms, more complex
- **Def:** An algorithm that chooses bid b_i^t for all $t = 1, \ldots, T$ is **"no-regret"** iff for any (adaptively chosen) $\left(b_{-i}^t\right)_{t=1,\ldots,T}$

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[u_i\left(b_i^t, b_{-i}^t\right)\right] \geq \max_{b^*}\left(\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[u_i\left(b^*, b_{-i}^t\right)\right]\right) - o(1)$$

- **Def:** An algorithm that chooses bid b_i^t for all $t = 1, \ldots, T$ is **"no-envy"** iff for any (adaptively chosen) $\left(b_{-i}^t\right)_{t=1,\ldots,T}$

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[u_i\left(b_i^t, b_{-i}^t\right)\right] \geq \max_{S^*}\left(v_i(S^*) - \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[p_{S^*,i}\left(b_{-i}^t\right)\right]\right) - o(1)$$

Threshold price to win set S^*

No-Envy vs No-Regret Learning

- Fix mechanism M
  - Suppose n bidders engage in a repeated execution of mechanism M
  - b_i^t: bidder i's action in round t
    - in SiSPAs, this is a vector of bids on each item
    - in more complex mechanisms, more complex
- **Def:** An algorithm that chooses bid b_i^t for all $t = 1, \dots, T$ is **"no-regret"** iff for any (adaptively chosen) $\left(b_{-i}^t\right)_{t=1,\dots,T}$

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[u_i(b_i^t, b_{-i}^t)] \geq \max_{b^*}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[u_i(b^*, b_{-i}^t)]\right) - o(1)$$

- **Def:** An algorithm that chooses bid b_i^t for all $t = 1, \dots, T$ is **"no-envy"** iff for any (adaptively chosen) $\left(b_{-i}^t\right)_{t=1,\dots,T}$

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[u_i(b_i^t, b_{-i}^t)] \geq \max_{S^*}\left(v_i(S^*) - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[p_{S^*,i}(b_{-i}^t)]\right) - o(1)$$

Threshold price to win set S^*

- **Provenance:** Walrasian equilibrium

No-Envy vs No-Regret Learning

- Fix mechanism M
  - Suppose n bidders engage in a repeated execution of mechanism M
  - b_i^t: bidder i's action in round t
    - in SiSPAs, this is a vector of bids on each item
    - in more complex mechanisms, more complex
- **Def:** An algorithm that chooses bid b_i^t for all $t = 1, \dots, T$ is **"no-regret"** iff for any (adaptively chosen) $\left(b_{-i}^t\right)_{t=1,\dots,T}$

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[u_i\left(b_i^t, b_{-i}^t\right)\right] \geq \max_{b^*}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[u_i\left(b^*, b_{-i}^t\right)\right]\right) - o(1)$$

- **Def:** An algorithm that chooses bid b_i^t for all $t = 1, \dots, T$ is **"no-envy"** iff for any (adaptively chosen) $\left(b_{-i}^t\right)_{t=1,\dots,T}$

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[u_i(b_i^t, b_{-i}^t)\right] \geq \max_{S^*}\left(v_i(S^*) - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[p_{S^*,i}\left(b_{-i}^t\right)\right]\right) - o(1)$$

Threshold price to win set S^*

- **Provenance:** Walrasian equilibrium

No-Envy vs No-Regret Learning

- Fix mechanism M
  - Suppose n bidders engage in a repeated execution of mechanism M
  - b_i^t: bidder i's action in round t
    - in SiSPAs, this is a vector of bids on each item
    - in more complex mechanisms, more complex
- **Def:** An algorithm that chooses bid b_i^t for all $t = 1, \dots, T$ is **"no-regret"** iff for any (adaptively chosen) $\left(b_{-i}^t\right)_{t=1,\dots,T}$

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[u_i\left(b_i^t, b_{-i}^t\right)\right] \geq \max_{b^*}\left(\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[u_i\left(b^*, b_{-i}^t\right)\right]\right) - o(1)$$

≥ in SiSPAs

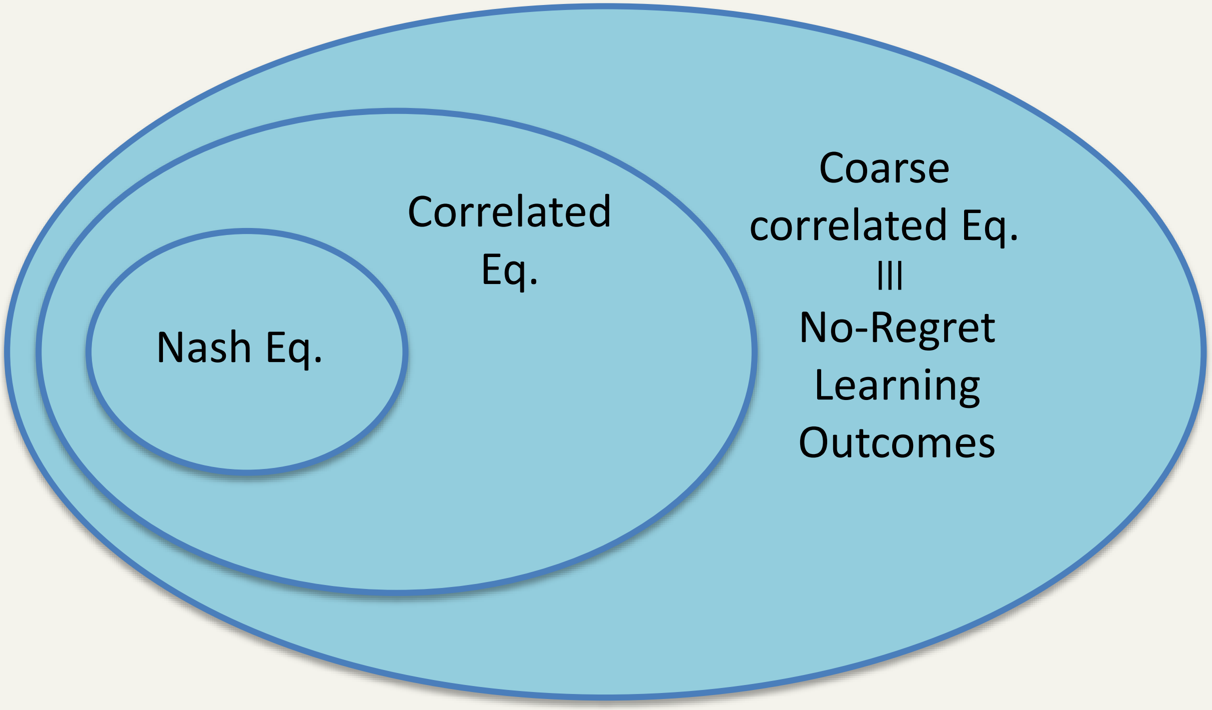
- **Def:** An algorithm that chooses bid b_i^t for all $t = 1, \dots, T$ is **"no-envy"** iff for any (adaptively chosen) $\left(b_{-i}^t\right)_{t=1,\dots,T}$

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[u_i\left(b_i^t, b_{-i}^t\right)\right] \geq \max_{S^*}\left(v_i(S^*) - \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[p_{S^*,i}\left(b_{-i}^t\right)\right]\right) - o(1)$$

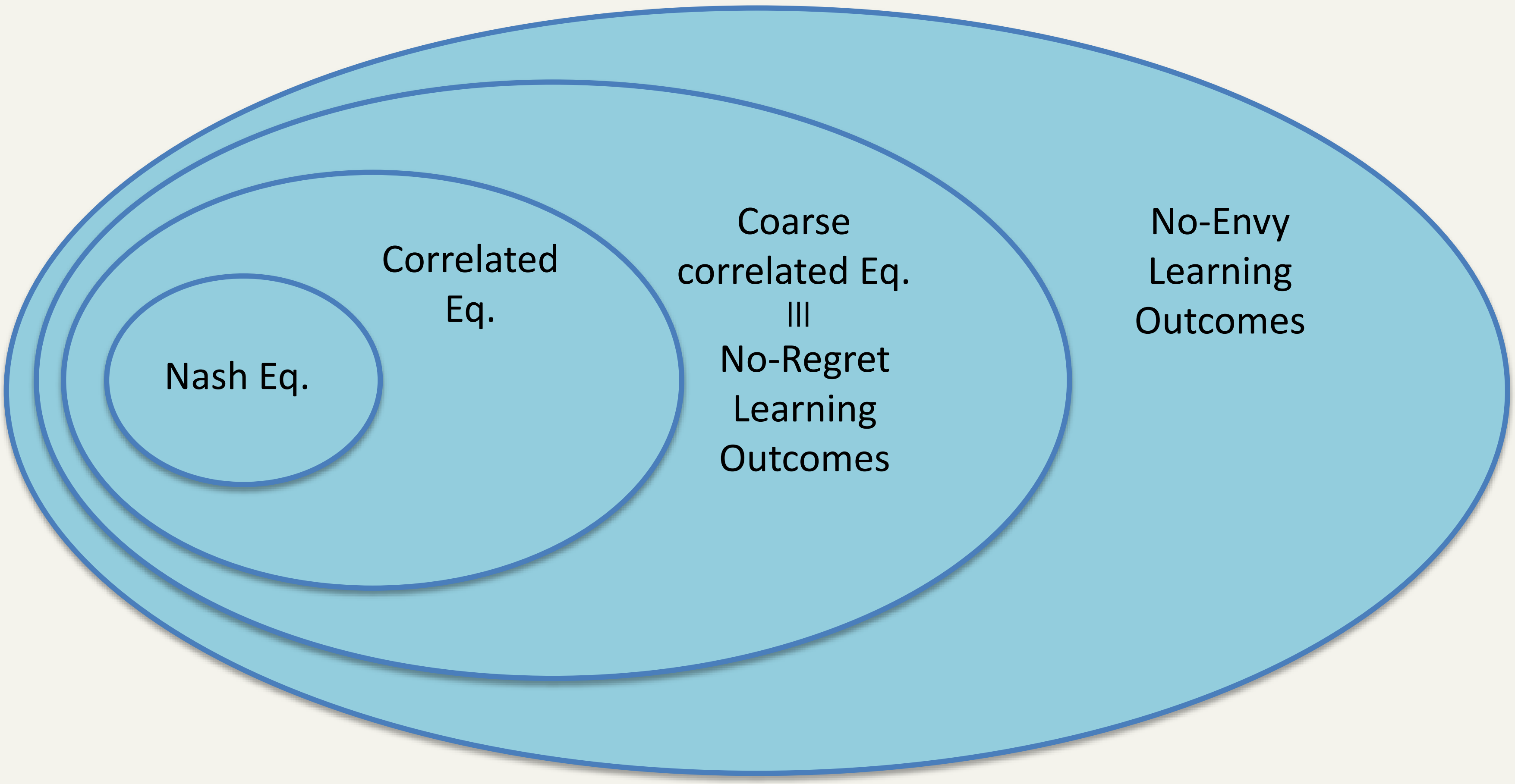
Threshold price to win set S^*

- **Provenance:** Walrasian equilibrium

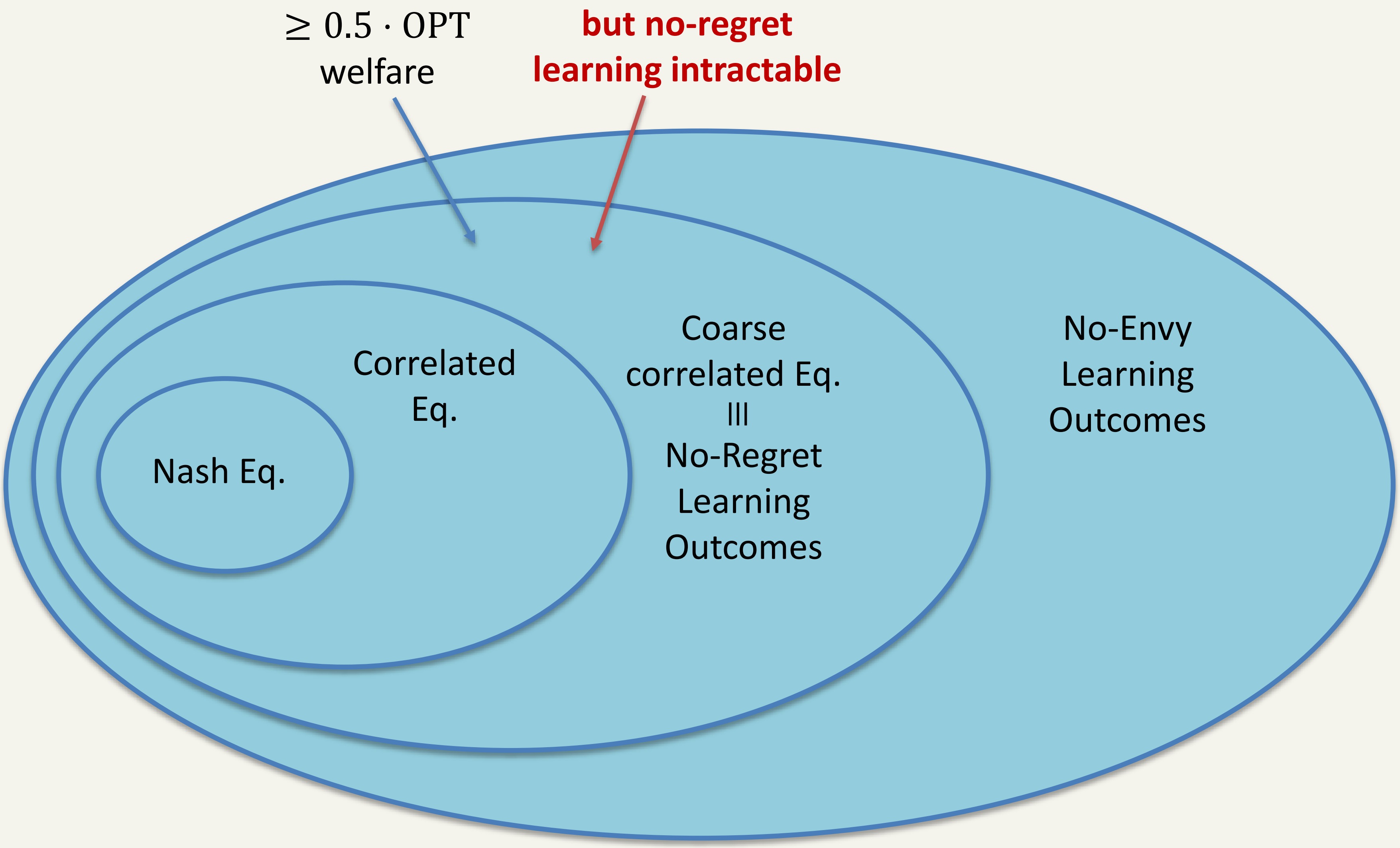
Solution Concepts in SiSPAs



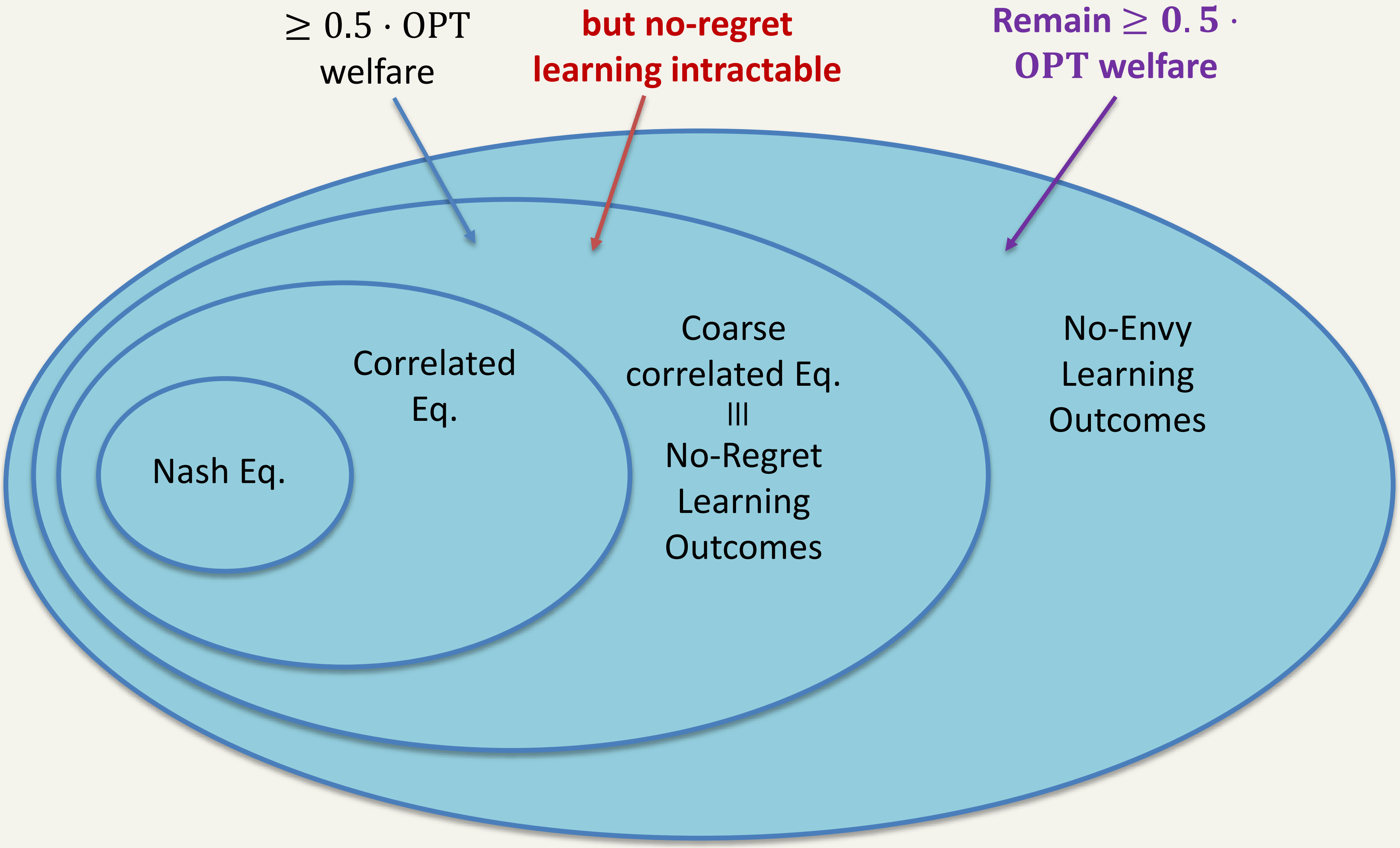
XOS bidders in SiSPAs



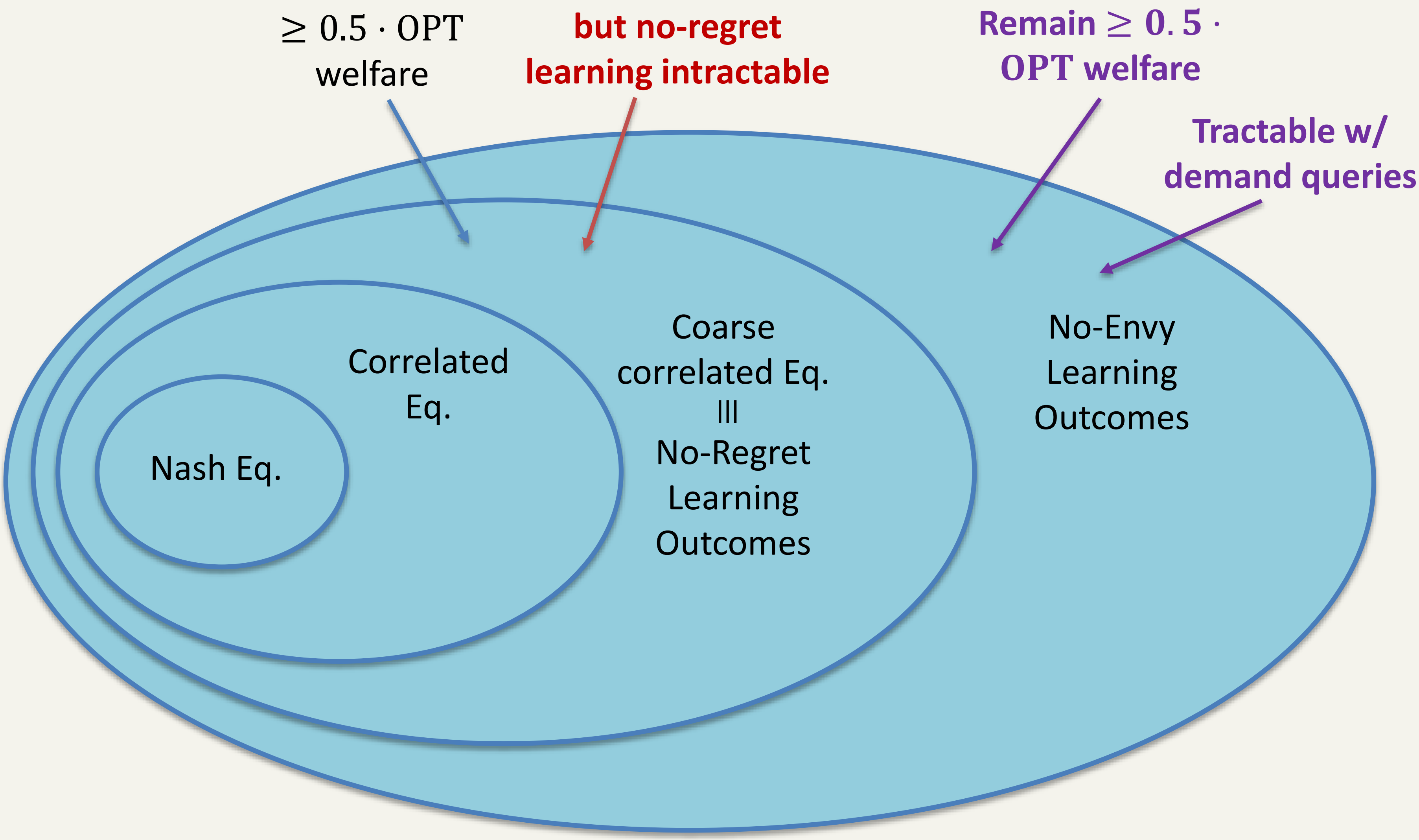
XOS bidders in SiSPAs



XOS bidders in SiSPAs



XOS bidders in SiSPAs

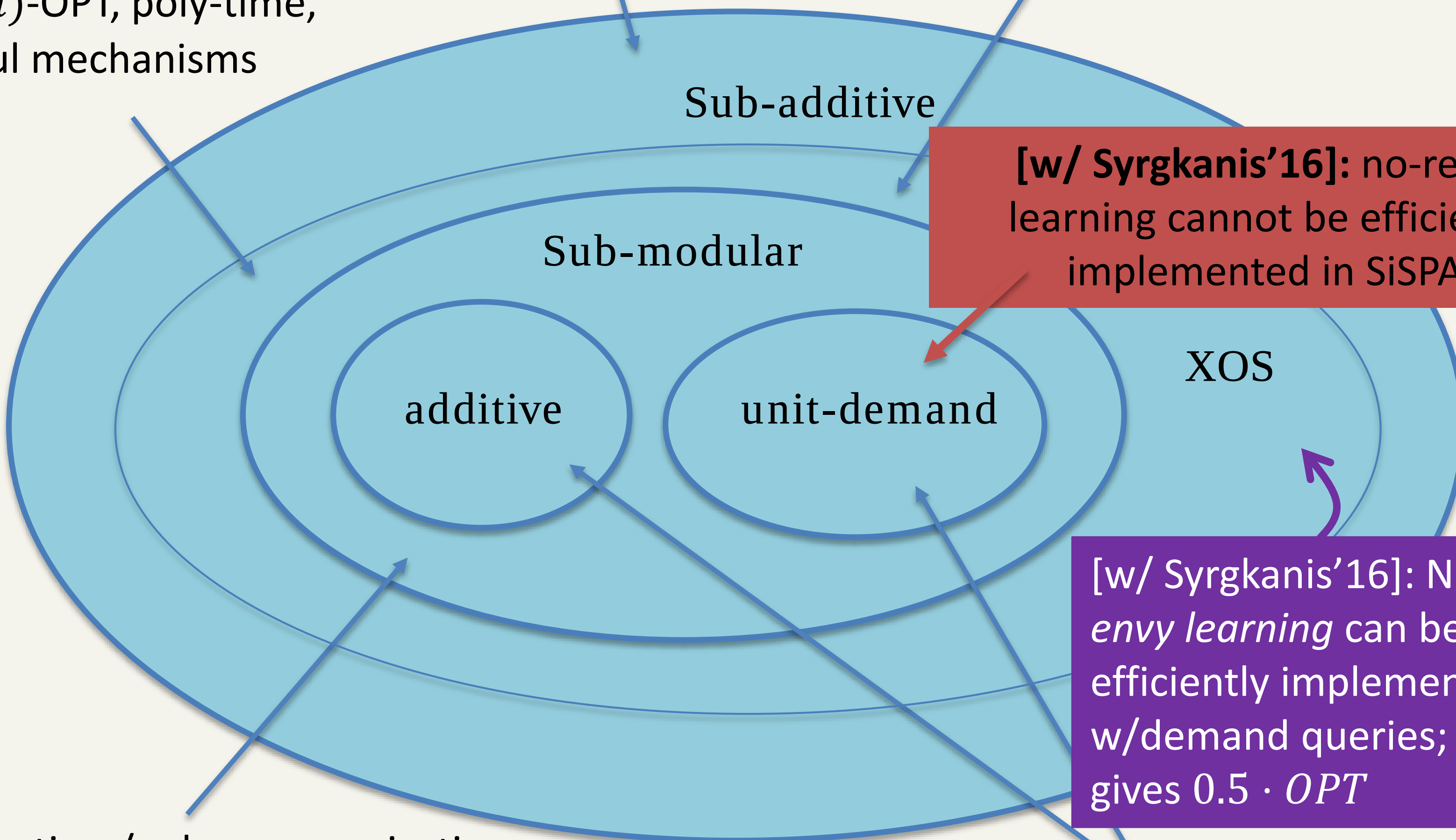


Welfare Optimization (evolving summary)

under no-regret learning, SiSPAs achieve 0.25-OPT

with *demand queries*, can get $O(\sqrt{\log m})$-OPT, poly-time, truthful mechanisms

demand queries + *Bayes:* 0.5-OPT, poly-time, truthful mechanisms



[w/ Syrgkanis'16]: no-regret learning cannot be efficiently implemented in SiSPAs

[w/ Syrgkanis'16]: N*o-envy learning* can be efficiently implemented w/demand queries; gives $0.5 \cdot OPT$

Truthful poly-time/poly-communication mechanisms lose polynomial factors

VCG gets OPT in poly-time, poly-communication and is truthful

The Menu

- **Combinatorial Auctions**
- **Truthfulness vs Computation vs Communication**
- **Beyond the Truthfulness Barrier**
- **Meantime in a More Practical Universe..**
- **Algorithmic Mechanism Design for Learning Agents**
- Discussion

The Menu

- Combinatorial Auctions
- Truthfulness vs Computation vs Communication
- Beyond the Truthfulness Barrier
- Meantime in a More Practical Universe..
- Algorithmic Mechanism Design for Learning Agents
- Discussion

Summary/Discussion

- Important practical applications call for a joint Economics and Computational approach to system engineering

- On the intellectual front, the pursuit can be condensed to the question:
  - "*How much more difficult are optimization problems on strategic input compared to honest input?*" **[Nisan-Ronen'99]**

Summary/Discussion

- Important practical applications call for a joint Economics and Computational approach to system engineering

- On the intellectual front, the pursuit can be condensed to the question:
  - "*How much more difficult are optimization problems on strategic input compared to honest input?*" **[Nisan-Ronen'99]**

- In Bayesian settings, the answer is "essentially not at all" **[Cai-Daskalakis-Weinberg'12-'15]**

Summary/Discussion

- Important practical applications call for a joint Economics and Computational approach to system engineering

- On the intellectual front, the pursuit can be condensed to the question:
  - *"How much more difficult are optimization problems on strategic input compared to honest input?"* **[Nisan-Ronen'99]**

- In Bayesian settings, the answer is "essentially not at all" **[Cai-Daskalakis-Weinberg'12-'15]**

- In non-Bayesian settings, intense research effort, but mostly negative results, even for the paradigmatic question of welfare optimization in combinatorial auctions [**Papadimitriou, Schapira, Singer'08; Buchfuhrer et al'10, Dughmi-Vondrak'11, Dobzinski'11,Dobzinski-Vondrak'12, Daniely, Schapira, Shahaf'15**]

Summary

- Approximation theory has shed light into the welfare guarantees of simple, non-truthful combinatorial auctions, such as simultaneous second price auctions (SiSPAs)

Summary

- Approximation theory has shed light into the welfare guarantees of simple, non-truthful combinatorial auctions, such as simultaneous second price auctions (SiSPAs)
  - constant factor approximations to OPT welfare, even for subadditive bidders, even when looking at no-regret learning outcomes
  - holds for a large class of smooth mechanisms

Summary

- Approximation theory has shed light into the welfare guarantees of simple, non-truthful combinatorial auctions, such as simultaneous second price auctions (SiSPAs)
  - constant factor approximations to OPT welfare, even for subadditive bidders, even when looking at no-regret learning outcomes
  - holds for a large class of smooth mechanisms

- We show that no-regret learning is intractable in SiSPAs, even for unit-demand bidders with the same value for all items, and who play against a stationary opponent

Summary

- Approximation theory has shed light into the welfare guarantees of simple, non-truthful combinatorial auctions, such as simultaneous second price auctions (SiSPAs)
  - constant factor approximations to OPT welfare, even for subadditive bidders, even when looking at no-regret learning outcomes
  - holds for a large class of smooth mechanisms

- We show that no-regret learning is intractable in SiSPAs, even for unit-demand bidders with the same value for all items, and who play against a stationary opponent

- We propose a different notion of learning, called *no-envy*

Summary

- Approximation theory has shed light into the welfare guarantees of simple, non-truthful combinatorial auctions, such as simultaneous second price auctions (SiSPAs)
  - constant factor approximations to OPT welfare, even for subadditive bidders, even when looking at no-regret learning outcomes
  - holds for a large class of smooth mechanisms
- We show that no-regret learning is intractable in SiSPAs, even for unit-demand bidders with the same value for all items, and who play against a stationary opponent
- We propose a different notion of learning, called *no-envy*
  - No-envy learning is efficiently implementable for XOS bidders in SiSPAS, using demand queries

Summary

- Approximation theory has shed light into the welfare guarantees of simple, non-truthful combinatorial auctions, such as simultaneous second price auctions (SiSPAs)
  - constant factor approximations to OPT welfare, even for subadditive bidders, even when looking at no-regret learning outcomes
  - holds for a large class of smooth mechanisms

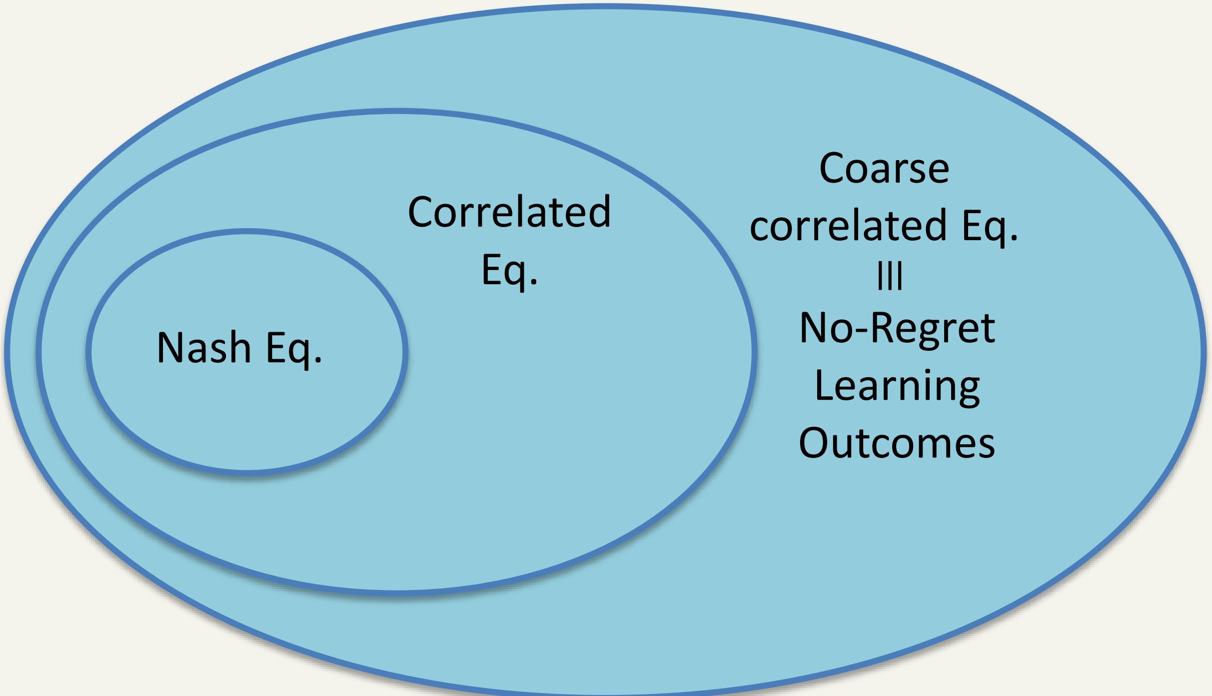
- We show that no-regret learning is intractable in SiSPAs, even for unit-demand bidders with the same value for all items, and who play against a stationary opponent

- We propose a different notion of learning, called *no-envy*
  - No-envy learning is efficiently implementable for XOS bidders in SiSPAS, using demand queries
  - No-envy learning outcomes is a larger set than no-regret outcomes

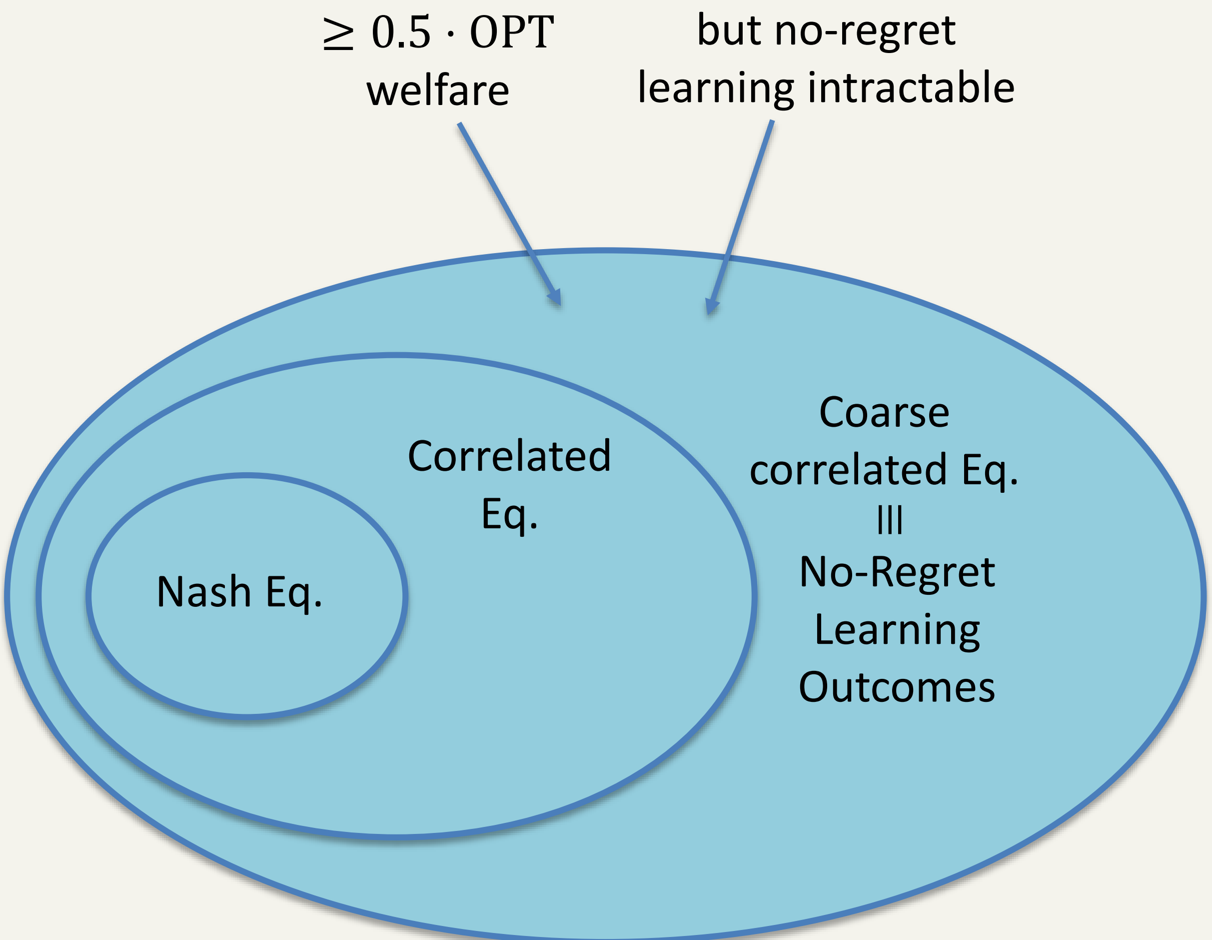
Summary

- Approximation theory has shed light into the welfare guarantees of simple, non-truthful combinatorial auctions, such as simultaneous second price auctions (SiSPAs)
  - constant factor approximations to OPT welfare, even for subadditive bidders, even when looking at no-regret learning outcomes
  - holds for a large class of smooth mechanisms
- We show that no-regret learning is intractable in SiSPAs, even for unit-demand bidders with the same value for all items, and who play against a stationary opponent
- We propose a different notion of learning, called *no-envy*
  - No-envy learning is efficiently implementable for XOS bidders in SiSPAS, using demand queries
  - No-envy learning outcomes is a larger set than no-regret outcomes
  - No-envy learning outcomes still guarantee half of optimal welfare

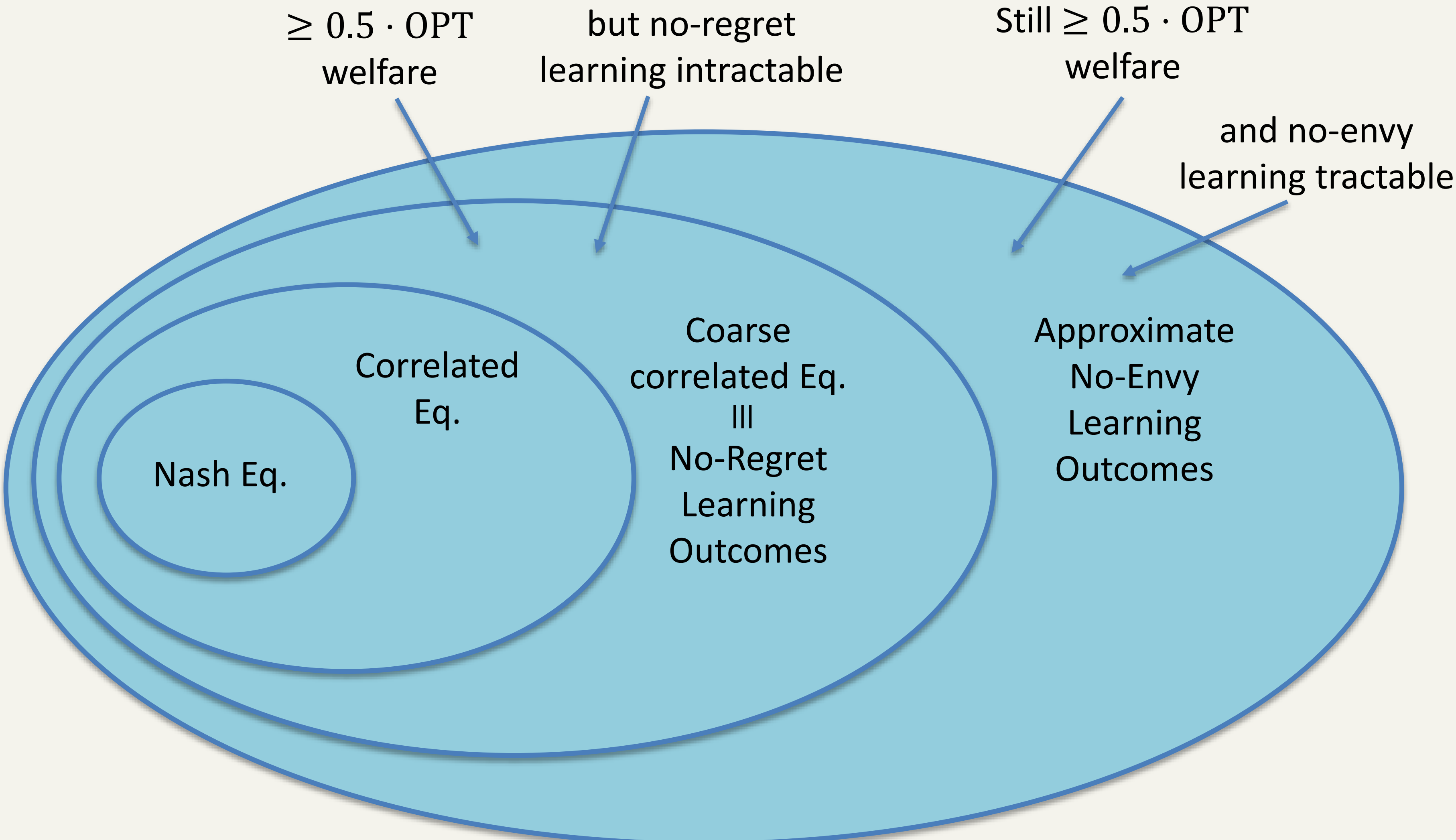
World view for XOS bidders in SiSPAs



World view for XOS bidders in SiSPAs



World view for XOS bidders in SiSPAs



Summary (cont'd)

- Approximation theory has shed light into the welfare guarantees of simple, non-truthful combinatorial auctions
- We show that no-regret learning is intractable in SiSPAs, even for unit-demand bidders who play against a stationary opponent
- We propose a different notion of learning, called *no-envy*, which is efficiently implementable for XOS bidders in SiSPAS, and discovers outcomes that approximately optimal
- Beyond SiSPAs, our upper bounds extend to a broad class of smooth mechanisms, including Simultaneous First Price and All-Pay auctions

Summary (cont'd)

- Approximation theory has shed light into the welfare guarantees of simple, non-truthful combinatorial auctions
- We show that no-regret learning is intractable in SiSPAs, even for unit-demand bidders who play against a stationary opponent
- We propose a different notion of learning, called *no-envy*, which is efficiently implementable for XOS bidders in SiSPAS, and discovers outcomes that approximately optimal
- Beyond SiSPAs, our upper bounds extend to a broad class of smooth mechanisms, including Simultaneous First Price and All-Pay auctions
- Open Problems:
  - go beyond XOS
  - go beyond welfare optimization
  - apply to practice, e.g. Uber, Urban Engines
  - interact with data

Summary (cont'd)

- Approximation theory has shed light into the welfare guarantees of simple, non-truthful combinatorial auctions
- We show that no-regret learning is intractable in SiSPAs, even for unit-demand bidders who play against a stationary opponent
- We propose a different notion of learning, called *no-envy*, which is efficiently implementable for XOS bidders in SiSPAS, and discovers outcomes that approximately optimal
- Beyond SiSPAs, our upper bounds extend to a broad class of smooth mechanisms, including Simultaneous First Price and All-Pay auctions
- Open Problems:
  - go beyond XOS
  - go beyond welfare optimization
  - apply to practice, e.g. Uber, Urban Engines
  - interact with data

Thanks!