Polynomial Sample Complexity for Blackbox Reductions in Mechanism Design with Additive Bidders

Arya Maheshwari

Princeton University

April 28, 2025

Mentors: Prof. Matt Weinberg, Eric Xue

Roadmap

1. Background
2. Existing procedure (Replica-Surrogate Matching)
3. Key obstacles toward polynomial sample complexity
4. Three main ideas for our solution

Background: Blackbox Reductions

This talk: welfare maximization setting.

What is a **blackbox reduction**?

- Input: m bidders (with types drawn from some distribution D), n items
- Given: some *algorithm* $\mathcal{A}$ achieving some welfare guarantee $\mathrm{Val}_{\mathcal{A}}(D)$
- **Goal**: some incentive-compatible *mechanism* $\mathcal{M}$ that achieves a similar welfare guarantee $\mathrm{Val}_{\mathcal{M}}(D)$ to $\mathrm{Val}_{\mathcal{A}}(D)$

"Incentive-compatible" means *Bayesian* incentive compatibility (**BIC**): optimal for a bidder to report truthfully *assuming* all other bidders do so.

Specific question: **sample complexity**:

Existing procedures require $\exp(n)$ samples from D. Can we achieve $O(\varepsilon)$-welfare apx. in $\mathrm{poly}(n, m, \frac{1}{\varepsilon})$ samples under *structured valuations*?

Background: Notation and Model

- n items, m (independent) bidders, 1 seller
- Bidder $k \in [m]$ type drawn from $D^{(k)}$ (assume $\text{supp}(D^{(k)}) \subseteq [0,1]^n$).
- Valuations: $v_k : \text{supp}(D^{(k)}) \times \{0,1\}^n \to \mathbb{R}_{\geq 0}$
- **Assumption 1: independent items**: $D^{(k)} = \times_{t \in [n]} D_t^{(k)}$
- **Assumption 2: additive valuations:** $v_k(t_k, x) = \sum_{i \in [n]} x_i \cdot (t_k)_i$
- Expected welfare: $\text{Val}_{\mathcal{A}}(D)$ from algorithm $\mathcal{A}$, $\text{Val}_{\mathcal{M}}(D)$ from mechanism $\mathcal{M}$

Main Result: Suppose $D = \times_{k \in [m]} D^{(k)}$ is a product distribution over m bidders with additive valuations over n independent items, with each $D^{(k)}$ satisfying a general "regularity" condition. Let $\mathcal{A}$ be any algorithm achieving expected welfare $\mathrm{Val}_{\mathcal{A}}(D)$. Then there exists an exactly-BIC mechanism $\mathcal{M}$ that achieves $\mathrm{Val}_{\mathcal{M}}(D) \geq \mathrm{Val}_{\mathcal{A}}(D) - O(\varepsilon)$ using at most $\mathrm{poly}(n, m, \frac{1}{\varepsilon})$ samples.

Extensions:

- (Forthcoming) Removing the regularity condition
- (Easy) Generalizing additive valuations to e.g. 1-Lipschitz
- (Hope) Extend objective to revenue maximization

Note: We focus on sample complexity. No claims about runtime.

Base Procedure: Replica-Surrogate Matching [HKM11]

- Goal: Given input algorithm $\mathcal{A}$ as a blackbox, create an exactly BIC mechanism $\mathcal{M}$ (with good welfare guarantee relative to $\mathcal{A}$) for multiple bidders.

- Plan: Turn the multi-bidder reduction problem into m separate single-bidder reduction problems

- Idea: Create a separate "interface layer" for each bidder that wraps around $\mathcal{A}$ in a way that (a) guarantees BIC while (b) ensuring small-enough objective (i.e. welfare) loss

Citation: Overview of replica-surrogate matching is drawn from prior survey in [MW24].

Base Procedure: Replica-Surrogate Matching (Continued)

Interface: **surrogate selection** procedure, run separately for each bidder k.

- Draw some number of ***surrogate*** types from $D^{(k)}$. ($\star$)
- Match bidder k to a surrogate that will be inputted to $\mathcal{A}$ in k's place.
- How to match: by drawing make-believe ***replica*** types from $D^{(k)}$, and then having bidder k "compete against" replicas for a surrogate.

Upshot: Competition with replicas is just a way to *induce prices* on surrogate types for each bidder in such a way that:

1. makes the new mechanism (based on adding this interface) BIC
2. approximately preserves welfare.

($\star$): In general, might have two different distributions: distribution D for $\mathcal{A}$ ($\Rightarrow$ surrogates from D) vs. input distribution D' for new mechanism ($\Rightarrow$ replicas from D').

Citation: Overview of replica-surrogate matching is drawn from prior survey in [MW24].

Replica-Surrogate Matching: Algorithm [HKM11, RW15]

Procedure: For each bidder k, run Phase 1. Then run Phase 2.

Phase 1: Surrogate Selection

1. Sample S values from $D^{(k)}$, and call these the *surrogates* $\mathrm{s} \in \mathcal{T}_k^S$.
2. Sample $S-1$ values from $D^{(k)}$; elicit k's type. Together, call these the *replicas* $\mathrm{r} \in \mathcal{T}_k^S$.
3. Create a complete bipartite graph G_k on vertices $V_k = \mathrm{r} \sqcup \mathrm{s}$.
   Weight v_{ij} of the $\mathrm{r}_i \leftrightarrow \mathrm{s}_j$ edge = "r_i's value for being matched to s_j"
   (in expectation over other bidders' draws from D_{-k}):
   $$v_{ij} = \mathop{\mathbb{E}}_{t_{-k}\sim D_{-k}}\left[\mathop{\mathbb{E}}_{o\sim\mathcal{A}(s_j;t_{-k})}[v_k(r_i,o)]\right]$$
4. Viewing edge weights as valuations of replicas ("buyers") for surrogates ("items"), run the VCG mechanism over matchings, i.e. compute the *maximum-weight* matching and corresponding VCG payments. *Note: this will be a perfect matching (non-negative edges).*

Phase 2: Surrogate Competition

1. For each bidder k, let b_k denote the surrogate that was matched to the replica representing bidder k in G_k.
2. Run $\mathcal{A}$ on input bid $\mathrm{b} = (b_1, \ldots, b_n)$. Let $o = (o_1, \ldots, o_n) = \mathcal{A}(\mathrm{b})$ be the resulting outcome.
3. Each bidder k gets o_k and pays the VCG payment for getting surrogate b_k in Phase 1.

Replica-Surrogate Matching: BIC Analysis

Claim (**BIC**): *if all bidders $j \neq k$ report truthfully, then bidder k is incentivized to report truthfully.* [HKM11, RW15]

- **“Stationarity”**: For any j, if bidder j reports truthfully to $\mathcal{M}$, then the *distribution of surrogate matched to bidder* j is precisely $D^{(j)}$.
  - Justification: via Principle of Deferred Decisions
  - Interpretation: adding the interface layer does *not* alter the distribution

- Implication: Assuming all bidders $j \neq k$ report truthfully, edge weights correctly captures bidder k's actual value for the surrogate.
  - Justification: recall that edge weights are computed in expectation over *draws* of all other bidders (i.e., assuming truthful reports).

- VCG pricing then means that it is optimal to report type truthfully.

Citation: Overview of replica-surrogate matching is drawn from prior survey in [MW24].

Replica-Surrogate Matching: Welfare Analysis

Welfare claim ($\mathrm{Val}_{\mathcal{M}}(D) \approx \mathrm{Val}_{\mathcal{A}}(D)$): more involved. [HKM11]

$\mathrm{Val}^k_{\mathcal{M}}(D)$: average "value of replica for surrogate it's matched to"
$\mathrm{Val}^k_{\mathcal{A}}(D)$: average "value of surrogate for itself"

Main ideas:

1. If a matching were to match a *large-enough* fraction of replicas to *close-enough* surrogate types, then it has *high-enough* weight to not lose that much welfare (i.e. $\mathrm{Val}_{\mathcal{M}}(D)$ close to $\mathrm{Val}_{\mathcal{A}}(D)$).
2. For large enough S (# of surrogates), the expected fraction of replicas that can be matched to close-enough surrogates is large-enough.
3. The maximum-weight matching is at least this good.

Replica-Surrogate Matching: Welfare Analysis

Welfare claim $(\mathrm{Val}_{\mathcal{M}}(D) \approx \mathrm{Val}_{\mathcal{A}}(D))$: more involved. [HKM11]

$\mathrm{Val}^k_{\mathcal{M}}(D)$: average "value of replica for surrogate it's matched to"
$\mathrm{Val}^k_{\mathcal{A}}(D)$: average "value of surrogate for itself"

Main ideas:

- If a matching were to match a *large-enough* fraction of replicas to *close-enough* surrogate types, then it has *high-enough* weight to not lose that much welfare (i.e. $\mathrm{Val}_{\mathcal{M}}(D)$ close to $\mathrm{Val}_{\mathcal{A}}(D)$).
- For large enough S (# of surrogates), the expected fraction of replicas that can be matched to close-enough surrogates is large-enough.
- The maximum-weight matching is at least this good.

Key Tension: Exponential Cover, Polynomial Samples?

How large does S need to be?

- Relates to an appropriate notion of *size of underlying type space.*
- For a type space like $[0, 1]^n$, this turns out to be **exponential** in n. :(

$\Rightarrow$ **Key tension**:

- Need exponentially many replicas/surrogates to run matching,
- Yet only want to take polynomially many samples from each $D^{(k)}$ ... ?

Our main result: designing a mechanism that achieves both of these desiderata.

Core Idea 1: Sampling by Products

Recall our assumption on valuations: *additive* over *independent items*.

Idea: leverage independence over items:

- Recall each bidder's distribution is $D^{(k)} = \times_{t \in [n]} D_t^{(k)}$.
- Alternate way to draw samples from $D^{(k)}$: draw samples $\mathcal{S}_t$ from each *marginal* $D_t^{(k)}$, then construct product set $\mathcal{S} := \times_{t \in [n]} \mathcal{S}_t$.

Caveat: values in $\mathcal{S}$ are *not* i.i.d. samples!

Issue: including bidder report before constructing products
$\Rightarrow$ bidder "influences" a fraction of replicas
$\Rightarrow$ bidder could manipulate surrogate prices by misreporting

Core Idea 2: Learning (Approximate) Surrogate Prices

Question: can we *decouple* learning good surrogate prices from the bidder's report?

Idea: *Two phases* of replica draws:

1. Draw *training replicas* (via products): do *not* include the bidder; learn correct prices for {training replicas}-{surrogates} matching
2. Draw *real replicas* (via products): *do* include bidder, and use prices from (1) in {real replicas}-{surrogates} matching.

Intuition: with enough samples, with high probability prices computed on training replica set will be *pretty good* for the real replica set.[1]

[1] Proof that formalizes this intuition is where additivity of valuations comes in.

Core Idea 2: Learning (Approximate) Surrogate Prices

Question: can we *decouple* learning good surrogate prices from the bidder's report?

Idea: *Two phases* of replica draws:

1. Draw *training replicas* (via products): do *not* include the bidder; learn correct prices for {training replicas}-{surrogates} matching
2. Draw *real replicas* (via products): *do* include bidder, and use prices from (1) in {real replicas}-{surrogates} matching.

Intuition: with enough samples, with high probability prices computed on training replica set will be *pretty good* for the real replica set.[1]

[1] Proof that formalizes this intuition is where additivity of valuations comes in.

Core Idea 3: Handling Small Errors and Failures

Issue: Two-phase learning procedure gives us *approximately correct* prices *most of the time*, but we need an *exactly* BIC mechanism.

Sources of "small errors"

1. *Approximately* correct prices $\Rightarrow$ surrogates slightly over/under-demanded (instead of perfect matching = even demand)
2. Inherent randomness in learning: small probability of getting "bad samples" $\Rightarrow$ prices not even approximately correct

Solution to (1): **Random dropping** and **dummy matching**

Solution to (2): **Discard *other* bidders upon sampling failure**

Core Idea 3: Handling Small Errors and Failures

Issue: small probabilities of bad sampling $\Rightarrow$ prices could be way off.

Solution : **Discard *other* bidders upon sampling failure**

- If detect bad sampling for bidder k, *all bidders $j \neq k$ get nothing.*
- Key: this preserves BIC property! because even if properties like stationarity now fail for bidder k, bidders $j \neq k$ *don't care*.

Consequences: Incurs additional welfare loss, but small-enough due to sampling failure probabilities being low-enough.

Summary

Three main ideas:

1. Construct exponentially-many replicas/surrogates from polynomially-many samples by taking **products**
2. **Two-phase** procedure of *training replicas* and *real replicas*
3. Resolve "**small errors**" from approximately correct prices and low-probability sampling failures

$\Rightarrow$ polynomial sample complexity for additive bidders over independent items.

Thank you! Questions?

Jason D. Hartline, Robert Kleinberg, and Azarakhsh Malekian.
Bayesian Incentive Compatibility via Matchings, pages 734–747.
2011.

Arya Maheshwari and Amanda Wang.
Survey of epsilon-bic to bic reductions with sample access via bernoulli factories.
Unpublished, 2024.

Aviad Rubinstein and S. Matthew Weinberg.
Simple mechanisms for a combinatorial buyer and applications to revenue monotonicity.
CoRR, abs/1501.07637, 2015.

Core Idea 3: Handling Small Errors and Failures

1. Surrogates slightly over/under-demanded
2. Small probabilities of bad sampling

Solution to (1): **Random dropping** and **dummy re-matching**

(i) Randomly drop matching edges with small probability, but large enough to remove all slight overdemanding

(ii) Add *dummy edges* to "fill" each surrogate to even demand

- Preserve incentives by (a) executing this agnostically to bidder report and (b) discarding any allocation obtained due to dummy edges.

Solution to (2): **Discard *other* bidders upon sampling failure**

- If detect bad sampling for bidder k, *all bidders* $j \neq k$ *get nothing.*
- Key: this preserves BIC property! because even if properties like stationarity now fail for bidder k, bidders $j \neq k$ *don't care*.

Consequences: both incur additional welfare loss, but small enough due to (1) low enough dropping and (2) sampling failure probabilities.