Limitations of Incentive Compatibility on Discrete Type Spaces

Taylor Lundy, Hu Fu

University of British Columbia

{tlundy, hufu}@cs.ubc.ca

Abstract

In the design of incentive compatible mechanisms, a common approach is to enforce incentive compatibility as constraints in programs that optimize over feasible mechanisms. Such constraints are often imposed on sparsified representations of the type spaces, such as their discretizations or samples, in order for the program to be manageable. In this work, we explore limitations of this approach, by studying whether all dominant strategy incentive compatible mechanisms on a set T of discrete types can be extended to the convex hull of T.

Dobzinski, Fu and Kleinberg (2015) answered the question affirmatively for all settings where types are single dimensional. It is not difficult to show that the same holds when the set of feasible outcomes is downward closed. In this work we show that the question has a negative answer for certain non-downward-closed settings with multi-dimensional types. This result should call for caution in the use of the said approach to enforcing incentive compatibility beyond single-dimensional preferences and downward closed feasible outcomes.

Introduction

Mechanism design studies optimization problems with private inputs from strategic agents. An agent's input, known as her *type*, is her private information including her valuation for the social outcomes, which are to be decided upon by the mechanism. A mechanism needs to solicit such information to achieve certain goals, e.g. maximizing welfare, revenue, surplus or fairness measures. It needs to provide its participants with correct incentives, via both social outcomes and payments, so that the agents find it in their best interests to reveal their types.

Dominant strategy incentive compatibility (DSIC) is one of the strongest and most widely used solution concepts that guarantee such incentives. Under DSIC, every participant, no matter what type she possesses and no matter what types the other participants report to the mechanism, will maximize her utility by revealing her true type. Not only is this a strong guarantee for the mechanism designer that true information should be reported and optimized over, it also alleviates the burden of strategizing from the participating agents — telling truth is a dominant strategy regardless of the other

agents' types or strategies. Partly thanks to this strong incentive guarantee, the two fundamental auctions, namely, the VCG auction that maximizes social welfare (Vickrey 1961; Clarke 1971; Groves 1973), and Myerson's auction that maximizes expected revenue for selling a single item (Myerson 1981), have been foundational in both the theory and practice of mechanism design.

As the scope of mechanism design expands beyond the classical settings, incentive compatible mechanisms that are optimal for various objective often lack the simple structures that characterize the VCG and Myerson's mechanisms. By and large, there have been two approaches to the design of incentive compatible mechanisms. The first approach focuses on classes of mechanisms that, by their simple structures, have obvious incentive guarantees. For example, in a multi-item auction, a sequential pricing mechanism puts prices on items and asks each agent in turn to choose her favorite items that remain; the bidders are not asked about their values, and choosing utility-maximizing items (according to their true values) is the obvious strategy to adopt (see, e.g., Chawla et al. 2010; Chawla, Malec, and Sivan 2010; Feldman, Gravin, and Lucier 2015). Another example is to optimize over parameterized "VCG-like" mechanisms which inherit incentive properties from the VCG mechanism (e.g. Sandholm and Likhodedov 2015). This approach is often used to search for mechanisms whose performance is a factor away from being optimal, since the optimal mechanism or its very close approximations are often not within the class of mechanisms being searched over.

The second approach forgoes structures that are easily interpretable, and exhaustively searches for the optimal mechanism. This is exemplified by solving mathematical programs (typically linear or convex programs) whose feasible regions contain the set of all incentive compatible mechanisms (see, e.g. Conitzer and Sandholm 2002; Dobzinski, Fu, and Kleinberg 2015; Daskalakis, Deckelbaum, and Tzamos 2017;Feng and Hartline 2018). Typically, incentive requirements are hardwired as constraints in such programs.

Difficulty arises in the second approach when one would like to adopt strong incentive guarantees such as DSIC, which need at least one constraint per profile of types to specify. When the space of possible types is a continuum, this gives rise to uncountably many constraints. While this does not always make the program impossible to solve it

considerably complicates the task. One way to work around this is to discretize the type space and only impose incentive compatible (IC) constraints on the set of discrete types used to represent the type space. Discretization is also embodied in the idea of a given prior distribution over a set of discrete types, on which the optimization can then be based (e.g. Conitzer and Sandholm 2002; Dobzinski, Fu, and Kleinberg 2015). The most common motivation for such prior distributions is that they naturally result from samples (e.g. from past observations or market research) from an underlying distribution, whereas the true distribution may be supported on a continuum of types. This approach motivates the question we study in this work.

Questions we study. In this work we aim to answer the question: when one has a mechanism that is DSIC on a discretized subset of a type space, can one always find a mechanism that has the same behavior on the subset and yet is DSIC on the whole type space? To make the question more concrete, we study the natural case where the whole type space is the convex hull of the discrete subset. To make the presentation easier, in the following we denote by T the discrete subset of types, and $\mathrm{Conv}(T)$ its convex hull.

We consider the question a fundamental one for the second approach to mechanism design that we described above. When one optimizes for a mechanism with IC constraints imposed only on T, if the resulting mechanism cannot be *extended* to the original type space, it loses incentive guarantees when put to use.

One objection may be that, given a mechanism that is DSIC on T, one may always run it on $\mathrm{Conv}(T)$, by restricting the "bidding language", so that a type in $\mathrm{Conv}(T)$ but not in T has to report a type in T. Such mechanisms, however, may lose the incentive guarantee which makes DSIC mechanisms attractive in the first place. Unless one can show that agents with types not in T have a dominant strategy in such mechanisms, such agents need to strategize over which types in T to report, depending on the types and strategies of their opponents. In the scenario where T is a set of samples from a continuous distribution, the vast majority of types may not be in T and have no incentive guarantee, which is clearly undesirable. In settings where the agents' types are single dimensional, Dobzinski, Fu, and Kleinberg (2015) showed that "restricting the bidding language" does turn any mechanism DSIC on T into a mechanism DSIC on any superset of T: each type in the superset has a dominant strategy, and by the revelation principle this gives rise to a DSIC mechanism that extends the given mechanism's behavior on T. To the best of our knowledge, no such guarantees are known beyond single dimensional settings.

Our Results. For agents with multi-dimensional types, we first give a condition under which any DSIC mechanism on T can be extended to a DSIC mechanism on $\mathrm{Conv}(T)$, via an argument that is different from Dobzinski, Fu, and Kleinberg's yet still straightforward (Theorem 2). In particular, the condition is satisfied whenever the set of feasible outcomes is *downward closed* (Theorem 1).

Our main result, however, is a construction of a set T of multi-dimensional types and a DSIC mechanism on it, for which we show that no DSIC mechanism on $\mathrm{Conv}(T)$ can output the same social outcomes on types in T. The impossibility result stands even if the extension mechanism is allowed to be randomized. This shows that, without conditions such as single-dimensional types or downward closed set of feasible outcomes, designing incentive compatible mechanisms by focusing on a discrete subset of types can be a questionable approach to designing mechanisms for the whole type space — there may not be any mechanism DSIC on the whole type space which behave the same way on the subset.

Near the end, we give a multi-dimensional setting where the expected *revenue* of a mechanism with only correct incentives for a set T of types can be unboundedly more than the revenue of a mechanism for $\mathrm{Conv}(T)$. This example is much less involved than our main result, because revenue optimal mechanisms are meaningful only when they do not overcharge any reported type and guarantee non-negative utility. This constraint can be much more stringent when imposed for all types in $\mathrm{Conv}(T)$ than for T only.

Related Works

For multi-dimensional preferences, an allocation rule for any fixed agent is implementable if and only if it satisfies the so-called *cyclic monotonicity* property (Rochet 1987). When the type space is convex, it turns out the weaker condition of *weak monotonicity* suffices for implementability (Saks and Yu 2005; Archer and Kleinberg 2014). It is notable that the two solution concepts we compare in Section precisely correspond to the case where the type space is convex and that where it is not. However, nowhere in our arguments do we make use of this beautiful fact.

Another closely related body of work, is the literature on automated mechanism design. In automated mechanism design, mechanisms are optimized for the setting and objective automatically using information about agents' type distributions. When this work was introduced by Conitzer and Sandholm (2002) the input for this problem was an explicit description of the agents distributions, however recent work has moved towards replacing this explicit description with samples from the agents type distribution (Likhodedov and Sandholm 2004; 2005; Sandholm and Likhodedov 2015). Our work highlights how interpolating between discrete samples can effect not only the objective but the implementablility of the mechanism itself. Luckily, the research on sample based automated mechanism design is able to avoid the pitfalls of only having discrete samples. They do this either by optimizing over parameterized families of mechanisms which are guaranteed to be implementable on the entire typespace or by working in settings where the addition of new types has no effect on the objective (i.e. downward closed settings, see Theorem 1)(Sandholm and Likhodedov 2015; Guo and Conitzer 2010; Balcan, Sandholm, and Vitercik 2018; Morgenstern and Roughgarden 2016). However, our work points out some difficulties that might arise if one wishes to take a more general, non-parameterized approach to automated mechanism design in

settings which are not downward closed.

There are a variety of well-studied settings that are not downward closed in which extending a type space to its the convex hull could cause problems. One commonly studied not downward closed setting arises from the job scheduling problem introduced in Nisan and Ronen (2001) and later built upon by Chawla et al. (2013) and Ashlagi, Dobzinski, and Lavi (2012). since in this problem every job must eventually be scheduled and the set of feasible solutions is not downward closed. Another example of a non-downward closed setting is one-sided matching markets in which every agent must be matched with exactly one good. An example of a one-sided matching market is the fair housing allocation studied by Abebe et al. (2019). Finally, the facility location problem from Devanur, Mihail, and Vazirani (2005) also not downward closed.

Preliminaries

We consider a setting with N agents where each agent i has a private type t_i from her type space $T_i \subseteq \mathbb{R}_+^m$. The type profile $\boldsymbol{t} = (t_1, \ldots, t_N)$ denotes the vector of all agents' types, from the joint type space $\boldsymbol{T} := \prod_i T_i$.

We adopt the standard shorthand notation to write $\boldsymbol{t}_{-i} := (t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n)$ from $\boldsymbol{T}_{-i} := \prod_{j \neq i} T_j$.

An outcome (or, interchangeably, allocation) for agent i lies in $\mathbb{R}_+^m$; for an outcome x_i, the agent with type t_i has value $\langle t_i, x_i \rangle = \sum_{j=1}^m t_{ij} x_{ij}$. A social outcome is denoted by a vector $(x_1, \ldots, x_N) \in \mathbb{R}_+^{mN}$. The set of all feasible social outcomes (or allocations) is denoted $\mathscr{F} \subseteq \mathbb{R}_+^{mN}$.

For example, in a single-item auction, $m = 1$, each $t_i \in \mathbb{R}_+$ represents agent i's value for the item, and $\mathscr{F} \subseteq \mathbb{R}_+^N$ is the all zero vector (representing not selling) and the N standard bases (each representing selling to a corresponding agent). As another example, in a m-unit auction with unit-demand buyers, $\mathscr{F} \subseteq \mathbb{R}_+^N$ is the set of all integral points in $\{(x_1, \ldots, x_N) \in \mathbb{R}_+^{mN} \mid \sum_{i=1}^N x_{ij} \leq 1, j = 1, 2, \ldots, m\}$.

Mechanisms. A (direct revelation) mechanism consists of an *allocation rule* $\boldsymbol{x} : \boldsymbol{T} \to \mathscr{F}$ and a *payment rule* $\boldsymbol{p} : \boldsymbol{T} \to \mathbb{R}_+^N$. The mechanism elicits type reports from the agents, and on reported type profile $\boldsymbol{t}$, decides on an allocation $\boldsymbol{x}(\boldsymbol{t}) \in \mathscr{F}$, with each agent i making a payment of $p_i(\boldsymbol{t})$. In general, allocation rules can be randomized, in which case $\boldsymbol{x}(\boldsymbol{t})$ is a randomized variable supported on $\mathscr{F}$. $\boldsymbol{x}(\cdot)$ induces allocation rule $x_i(\cdot)$ for each agent i: for all $\boldsymbol{t}$, $x_i(\boldsymbol{t}) \in \mathbb{R}_+^m$ is the vector consisting of the $[(i-1)m+1]$-st to the im-th coordinates in $\boldsymbol{x}(\boldsymbol{t})$. When $\boldsymbol{x}(\boldsymbol{t})$ is a random variable, so are $x_i(\boldsymbol{t})$'s.

When $\boldsymbol{x}(\boldsymbol{t})$ is deterministic, we write $\boldsymbol{x}(\boldsymbol{t}) = \mathbf{y} \in \mathscr{F}$ as a shorthand for $\mathbf{Pr}[\boldsymbol{x}(\boldsymbol{t}) = \mathbf{y}] = 1$.

Agents have quasi-linear utilities, that is, when reporting type t_i', agent i's utility is $\mathbf{E}[\langle t_i, x_i(t_i', \boldsymbol{t}_{-i}) \rangle] - p_i(t_i', \boldsymbol{t}_{-i})$ (where the expectation is taken over the randomness in $\boldsymbol{x}(\boldsymbol{t})$).

A mechanism is *dominant strategy incentive compatible* (DSIC) if, for all $\boldsymbol{t} \in \boldsymbol{T}$, and for all $t_i' \in T_i$, $\mathbf{E}[\langle t_i, x_i(t_i, \boldsymbol{t}_{-i}) \rangle] - p_i(t_i, \boldsymbol{t}_{-i}) \geq \mathbf{E}[\langle t_i, x_i(t_i', \boldsymbol{t}_{-i}) \rangle] -$ $p_i(t_i', \boldsymbol{t}_{-i})$. An allocation rule $\boldsymbol{x}$ is said to be DSIC implementable or simply DSIC if there is a payment rule $\boldsymbol{p}$ such that $(\boldsymbol{x}, \boldsymbol{p})$ is a DSIC mechanism. In this case, we say $\boldsymbol{x}$ is implemented by payment rule $\boldsymbol{p}$.

Extensions. Given a subset $S \subseteq \mathbb{R}^n$, we denote by $\mathrm{Conv}(S)$ the convex hull of S.

Definition 1. *An allocation rule $\tilde{\boldsymbol{x}} : \mathrm{Conv}(\boldsymbol{T}) \to \mathscr{F}$ is an* extension *of an allocation rule $\boldsymbol{x} : \boldsymbol{T} \to \mathscr{F}$ if for all $\boldsymbol{t} \in \boldsymbol{T}$, $\tilde{\boldsymbol{x}}(\boldsymbol{t})$ has the same distribution as $\boldsymbol{x}(\boldsymbol{t})$. Similarly, a payment rule $\tilde{\boldsymbol{p}} : \mathrm{Conv}(\boldsymbol{T}) \to \mathbb{R}_+^N$ is an extension of payment rule $\boldsymbol{p} : \boldsymbol{T} \to \mathbb{R}_+^N$ if for all $\boldsymbol{t} \in \boldsymbol{T}$, $\tilde{\boldsymbol{p}}(\boldsymbol{t}) = \boldsymbol{p}(\boldsymbol{t})$.*

In Definition 1, if $\boldsymbol{x}(\cdot)$ is deterministic, then $\tilde{\boldsymbol{x}}(\cdot)$ being an extension simply means $\tilde{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{x}(\boldsymbol{t})$ for all $\boldsymbol{t} \in \boldsymbol{T}$.

Downward closed settings. The feasible allocation set $\mathscr{F}$ is *downward closed* if $\mathbf{y} \in \mathscr{F}$ entails $\boldsymbol{x} \in \mathscr{F}$ for all $\boldsymbol{x} \preceq \mathbf{y}$, where $\boldsymbol{x} \preceq \mathbf{y}$ denotes $x_j \leq y_j$ for $j = 1, \cdots, mN$.

Weak monotonicity. A well-known necessary condition for an allocation rule to be DSIC implementable is weak monotonicity:

Definition 2. *An allocation rule $\boldsymbol{x} : \boldsymbol{T} \to \mathscr{F}$ is weakly monotone if for each agent i, any $t_i, t_i' \in T_i$ and $\boldsymbol{t}_{-i} \in \boldsymbol{T}_{-i}$,*

$$\mathbf{E}\left[\langle t_i - t_i', x_i(t_i, \boldsymbol{t}_{-i}) - x_i(t_i', \boldsymbol{t}_{-i}) \rangle\right] \geq 0.$$

Proposition 1 (see, e.g., Saks and Yu 2005). *An allocation rule is implementable only if it is weakly monotone.*

In fact, Saks and Yu (2005) showed that, if $\boldsymbol{T}$ is convex, then weak monotonicity is also a sufficient condition for DSIC implementability.

Revenue. A mechanism $(\boldsymbol{x}, \boldsymbol{p})$ is ex post *individually rational* (IR) if for each agent i and for every $\boldsymbol{t} \in \boldsymbol{T}$, $\langle t_i, x_i(\boldsymbol{t}) \rangle - p_i(\boldsymbol{t}) \geq 0$.

Given a distribution D on $\boldsymbol{T}$ and a mechanism $(\boldsymbol{x}, \boldsymbol{p})$ that is DSIC and ex post IR, the *expected revenue* of the mechanism is $\mathbf{E}_{\boldsymbol{t} \sim D}[\sum_i p_i(\boldsymbol{t})]$. The *optimal* revenue is the maximum expected revenue achievable among all DSIC, ex post IR mechanisms.

DSIC Convex Extensions

Before presenting our main result on the impossibility to extend DSIC allocation rules, we first complement Dobzinski, Fu, and Kleinberg (2015)'s result in single-dimensional setting with a simple observation in multi-dimensional preference settings: whenever the feasible allocation space is downward closed, any DSIC allocation rule on a type space can be extended to its convex hull by another DSIC allocation rule.

Theorem 1. *If the set of feasible allocations $\mathscr{F}$ is downward closed, for any DSIC allocation rule $\boldsymbol{x}$ on a type space $\boldsymbol{T}$, there is a DSIC extension $\tilde{\boldsymbol{x}}$ of $\boldsymbol{x}$ on $\mathrm{Conv}(\boldsymbol{T})$. If $\boldsymbol{x}$ is implemented with a payment rule $\boldsymbol{p}$, $\tilde{\boldsymbol{x}}$ can be implemented by an extension $\tilde{\boldsymbol{p}}$ of $\boldsymbol{p}$. If $\boldsymbol{p}$ is individually rational on $\boldsymbol{T}$, so is $\tilde{\boldsymbol{p}}$ on $\mathrm{Conv}(\boldsymbol{T})$.*

If we do not require the statement about individual rationality, extensibility is guaranteed by an even weaker condition, which we call *single swap feasible*.

Definition 3 (Single swap feasible). *A feasible allocation set $\mathscr{F}$ is* single swap feasible *(SSF) if for every agent i there exists an allocation $\boldsymbol{x}^{\mathrm{ssf}}(i) \in \mathscr{F}$ such that for any $\boldsymbol{x}' \in \mathscr{F}$, $(x_i', \boldsymbol{x}_{-i}^{\mathrm{ssf}}(i)) \in \mathscr{F}$.*

Intuitively, $\boldsymbol{x}^{\mathrm{ssf}}(i)$ is a feasible allocation vector such that if we replace the ith element of this vector with the ith element from any other feasible allocation the resulting allocation is still feasible. If $\mathscr{F}$ is a product space or is downward closed, it must be SSF. [1]

Theorem 2. *If the set of feasible allocations $\mathscr{F}$ is SSF, any DSIC allocation rule $\boldsymbol{x}$ on a type space $\boldsymbol{T}$, there is a DSIC extension $\tilde{\boldsymbol{x}}$ of $\boldsymbol{x}$ on $\mathrm{Conv}(\boldsymbol{T})$.*

The proofs for both Theorem 1 and Theorem 2 can be found in the supplementary materials of the full paper. The main result of this paper is that without this condition a DSIC extension may not exist.

Theorem 3. *There is a two agent type space $\boldsymbol{T}^{\mathrm{rand}}$ with a DSIC allocation rule $\boldsymbol{x}^{\mathrm{rand}}$, such that $\boldsymbol{x}^{\mathrm{rand}}$ cannot be extended by a DSIC allocation rule to $\mathrm{Conv}(\boldsymbol{T}^{\mathrm{rand}})$.*

We prove the theorem in two steps. We first present a setting with three-dimensional preferences for which we show the non-existence of *deterministic* extensions. We then build on the construction, lifting it to a higher dimension, where we strengthen the argument and show the non-existence of extensions that even allow randomization.

Non-existence of deterministic extensions

We first present type space $\boldsymbol{T}^{\mathrm{det}} = T_1^{\mathrm{det}} \times T_2^{\mathrm{det}}$ and the allocation rule $\boldsymbol{x}^{\mathrm{det}}$, and then show that $\boldsymbol{x}^{\mathrm{det}}$ is DSIC and yet cannot be extended by any deterministic DSIC allocation rule on $\mathrm{Conv}(\boldsymbol{T}^{\mathrm{det}})$.

The two agents have identical type spaces: for $i = 1, 2$, $T_i^{\mathrm{det}} = T^{\mathrm{det}} := \{A = [1, 0, 0], B = [0, 1, 0], C = [0, 0, 1], D = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]\}$.

The allocation rule $\boldsymbol{x}^{\mathrm{det}}$ is also symmetric, in the sense that $x_1^{\mathrm{det}}(t_1, t_2) = x_2^{\mathrm{det}}(t_2, t_1)$ for any $t_1, t_2 \in T^{\mathrm{det}}$. We summarize x_1^{det} with the diagram below. The rows are indexed by agent 1's own type V_1, and the columns by agent 2's type V_2:

$$
\begin{array}{c} \\ A \\ B \\ C \\ D \end{array}
\begin{array}{cccc}
A & B & C & D \\
\left[\begin{matrix} [1,1,0] & [2,0,2] & [3,0,3] & [4,0,4] \\ [0,1,1] & [2,2,0] & [3,3,0] & [4,4,0] \\ [1,0,1] & [0,2,2] & [0,3,3] & [0,4,4] \\ [0,1,1] & [2,2,0] & [3,3,0] & [4,4,0] \end{matrix}\right]
\end{array}
$$

The set of all feasible allocations is then $\mathscr{F} = \{(x_1^{\mathrm{det}}(V_1, V_2), x_2^{\mathrm{det}}(V_1, V_2))\}_{V_1, V_2 \in T^{\mathrm{det}}}$. We hasten to point out that $\mathscr{F}$ is *not* the product between the two agents' respective set of feasible allocations. For example, $[1, 1, 0, 1, 1, 0]$ is in $\mathscr{F}$ as it is $\boldsymbol{x}^{\mathrm{det}}(A, A)$, but $[1, 1, 0, 0, 1, 1]$ is not. This is important for the proof.

---

[1]To see that a downward closed $\mathscr{F}$ is SSF, observe that we can let $\boldsymbol{x}^{\mathrm{ssf}}(i)$ be the all zero vector for each agent i.

Proposition 2. $\boldsymbol{x}^{\mathrm{det}}$ *is DSIC implementable.*

Proof. Let the payment be 0 for both agents and all type profiles. As the allocation and payment rules are both symmetric, consider either agent i. If $t_{-i} = A$, the maximum value agent i could get, when her type is A, B, or C, is 1, attained with truthful bidding. For $t_i = D$, the four allocations all give the same value $\frac{2}{3}$. Similar arguments hold when t_{-i} is B, C or D. □

Theorem 4. *There exists no deterministic DSIC extension of $\boldsymbol{x}^{\mathrm{det}}$.*

Before proving Theorem 4, we make several preparatory observations.

A key difficulty with multi-dimensional preferences is the the lack of a payment identity à la Myerson (Myerson 1981). In order to argue that any extension of an allocation rule is not DSIC, one has to either check the many cyclic (or weak) monotonicity conditions, or show that no payment rule can support the extension in a DSIC mechanism. We designed $\boldsymbol{T}^{\mathrm{det}}$ and $\boldsymbol{x}^{\mathrm{det}}$ carefully so that the allocations "lock" the payment rules.

Lemma 1. *For any allocation rule $\tilde{\boldsymbol{x}}$ that is an extension of $\boldsymbol{x}^{\mathrm{det}}$, if $\tilde{\boldsymbol{x}}$ can be implemented by a DSIC mechanism with payment rule $\boldsymbol{p}$, then for any $V \in \{A, B, C, D\}$, $p_1(A, V) = p_1(B, V) = p_1(C, V) = p_1(D, V)$, and $p_2(V, A) = p_2(V, B) = p_2(V, C) = p_2(V, D)$.*

Proof. We prove the lemma for agent 1, and the statement for agent 2 follows by symmetry.

By DSIC, for any $V \in T^{\mathrm{det}}$, we have

$$
\begin{aligned}
\langle A, x_1^{\mathrm{det}}(A, V) \rangle - p_1(A, V) \\
\geq \langle A, x_1^{\mathrm{det}}(C, V) \rangle - p_1(C, V); \\
\langle B, x_1^{\mathrm{det}}(B, V) \rangle - p_1(B, V) \\
\geq \langle B, x_1^{\mathrm{det}}(A, V) \rangle - p_1(A, V); \\
\langle C, x_1^{\mathrm{det}}(C, V) \rangle - p_1(C, V) \\
\geq \langle C, x_1^{\mathrm{det}}(B, V) \rangle - p_1(B, V).
\end{aligned}
$$

Note that

$$
\begin{aligned}
\langle A, x_1^{\mathrm{det}}(A, V) \rangle &= \langle A, x_1^{\mathrm{det}}(C, V) \rangle; \\
\langle C, x_1^{\mathrm{det}}(C, V) \rangle &= \langle C, x_1^{\mathrm{det}}(B, V) \rangle; \\
\langle B, x_1^{\mathrm{det}}(B, V) \rangle &= \langle B, x_1^{\mathrm{det}}(A, V) \rangle.
\end{aligned}
$$

Therefore $p_1(A, V) \leq p_1(C, V) \leq p_1(B, V) \leq p_1(A, V)$. Hence all inequalities are tight and we have $p_1(C, V) = p_1(A, V) = p_1(B, V)$.

For type D's payment, note that $x_1^{\mathrm{det}}(B, V) = x_1^{\mathrm{det}}(D, V)$. If $p_1(B, V) \neq p_1(D, V)$, one of B and D must be incentivized to misreport the other type. Therefore $p_1(B, V) = p_1(D, V)$. □

Lemma 2. *If $\tilde{x}$ is a deterministic DSIC extension of $\boldsymbol{x}^{\mathrm{det}}$, then for $t_1 = \frac{1}{3}A + \frac{1}{3}B + \frac{1}{3}D$, and any $V_2 \in T^{\mathrm{det}}$, $\tilde{x}_1(t_1, V_2) \in \{\tilde{x}_1(A, V_2), \tilde{x}_1(B, V_2), \tilde{x}_1(C, V_2), \tilde{x}_1(D, V_2)\}$.*

Proof. For the sake of contradiction, assume $\tilde{x}$ is a DSIC extension of x^{det}, implementable by payment rule p, and for t_1 and $V_2 \in T^{\mathrm{det}}$, $\tilde{x}_1(t_1, V_2) = x_1^{\mathrm{det}}(V_1, V_2')$ for some $(V_1, V_2') \in T^{\mathrm{det}}$ and $V_2' \neq V_2$.

For any V_2, one of $x_1^{\mathrm{det}}(A, V_2)$ and $x_1^{\mathrm{det}}(B, V_2)$ gives an equal positive value to both A and B. Let V_1^* be the type that induces this equally valued allocation. (For example, if $V_2 = B$, then $x_1^{\mathrm{det}}(A, B) = [2, 0, 2]$ and $x_1^{\mathrm{det}}(B, B) = [2, 2, 0]$. Both A and B have the same value for $x_1^{\mathrm{det}}(B, B)$ and so type B would be V_1^*.)

Observe that, for the allocation $x_1^{\mathrm{det}}(V_1, V_2')$, V_1 has positive value $\langle V_1, x_1^{\mathrm{det}}(V_1, V_2') \rangle$, and no other type has higher value for it. Therefore, type t_1 has value at most $\langle \frac{2}{3}V_1 + \frac{1}{3}D, x_1^{\mathrm{det}}(V_1, V_2') \rangle$ for the allocation. In order for t_1 to have no incentive to misreport V_1^*, we must have

$$\langle \frac{2}{3}V_1 + \frac{1}{3}D, x_1^{\mathrm{det}}(V_1, V_2') \rangle - p_1(t_1, V_2)$$
$$\geq \langle \frac{2}{3}V_1^* + \frac{1}{3}D, x_1^{\mathrm{det}}(V_1^*, V_2) \rangle - p_1(V_1^*, V_2) \quad (1)$$

On the other hand, in order for type V_1 not to have incentive for deviating to t_1, we have

$$\langle V_1, x_1^{\mathrm{det}}(V_1, V_2) \rangle - p_1(V_1, V_2)$$
$$= \langle V_1^*, x_1^{\mathrm{det}}(V_1^*, V_2) \rangle - p_1(V_1^*, V_2)$$
$$\geq \langle V_1, x_1^{\mathrm{det}}(V_1, V_2') \rangle - p_1(t_1, V_2); \quad (2)$$

where for the equality we used the fact that the value obtained by reporting truthfully is the same for every type in $\{A, B, C\}$ given a fixed type of the opponent, and that $p_1(V_1, V_2) = p_1(V_1^*, V_2)$ by Lemma 1.

Similarly, in order for type D not to have incentive for deviating to t_1, we have

$$\langle D, x_1^{\mathrm{det}}(D, V_2) \rangle - p_1(D, V_2)$$
$$= \langle D, x_1^{\mathrm{det}}(V_1^*, V_2) \rangle - p_1(V_1^*, V_2)$$
$$\geq \langle D, x_1^{\mathrm{det}}(V_1, V_2') \rangle - p_1(t_1, V_2), \quad (3)$$

where for the equality we used the fact that type D has the same value for all allocations given a fixed type of the opponent, and that $p_1(D, V_2) = p_1(V_1^*, V_2)$ by Lemma 1. Crucially, (2) and (3) cannot both be tight, because by construction, for any $V_2 \neq V_2'$,

$$\langle V_1, x_1^{\mathrm{det}}(V_1, V_2) - x_1^{\mathrm{det}}(V_1, V_2') \rangle$$
$$= \frac{3}{2} \langle D, x_1^{\mathrm{det}}(D, V_2) - x_1^{\mathrm{det}}(V_1, V_2') \rangle \neq 0.$$

Therefore, $\frac{2}{3} \cdot$ (2) $+ \frac{1}{3} \cdot$ (3) gives

$$\langle \frac{2}{3}V_1^* + \frac{1}{3}D, x_1^{\mathrm{det}}(V_1^*, V_2) \rangle - p_1(V_1^*, V_2)$$
$$> \langle \frac{2}{3}V_1 + \frac{1}{3}D, x_1^{\mathrm{det}}(V_1, V_2') \rangle - p_1(t_1, V_2),$$

which contradicts (1). $\qquad\square$

By the same reasoning as for Lemma 1, the following lemma follows from Lemma 2.

Lemma 3. *If $\tilde{x}$ is a deterministic DSIC extension of x^{det}, implementable by payment rule p, then for any t_1 in the interior of $\mathrm{Conv}(T^{\mathrm{det}})$ and any $V_2 \in T^{\mathrm{det}}$, $p_1(t_1, V_2) = p_1(A, V_2)$.*

We are now ready to prove Theorem 4.

Proof of Theorem 4. Suppose $\tilde{x}$ is a deterministic DSIC extension of x^{det}. We show a contradiction by showing that $\tilde{x}$ must violate weak monotonicity.

Consider $t_1 = \frac{1}{3}A + \frac{1}{3}B + \frac{1}{3}D$ and when agent 2's type is A. Since t_1 could report any type in T^{det}, she has as options $x_1^{\mathrm{det}}(A, A), x_1^{\mathrm{det}}(B, A), x_1^{\mathrm{det}}(C, A)$ and $x_1^{\mathrm{det}}(D, A)\}$, all at the same price by Lemma 3. By Lemma 2, these are also all the allocations she could possibly get.

Since $[1, 1, 0]$ is the only allocation for which both types A and B have positive value, it is t_1's preferred allocation, i.e., $\tilde{x}_1(t_1, A)$ must be $[1, 1, 0]$. This in turn implies $\tilde{x}_2(t_1, A) = [1, 1, 0]$. (Recall that $\mathscr{F}$ is not a product space, and the only allocation in which agent 1 gets $[1, 1, 0]$ is $x^{\mathrm{det}}(A, A) = [1, 1, 0, 1, 1, 0]$.)

Similarly, one can show $\tilde{x}_1(t_1, D) = [4, 4, 0]$, which implies $\tilde{x}_2(t_1, D) = [2, 2, 0]$ or $[4, 4, 0]$. But in either case, weak monotonicity is violated for agent 2's types A and D. For example, if $\tilde{x}_2(t_1, D) = [2, 2, 0]$, we have

$$\langle A, [1, 1, 0] \rangle + \langle D, [2, 2, 0] \rangle < \langle A, [2, 2, 0] \rangle + \langle D, [1, 1, 0] \rangle.$$

Therefore, no deterministic DSIC extension of x^{det} is possible. $\qquad\square$

Non-existence of randomized extensions

We need a more convoluted construction and a more careful argument to prove the impossibility of extensions that are possibly randomized. We build on T^{det} and x^{det} to construct T^{rand} and x^{rand} and prove Theorem 3.

We first raise types in T^{det} to a space of seven dimensions. Define $A' = [1, 0, 0, 0, 0, 0, 0]$, $B' = [0, 1, 0, 0, 0, 0, 0]$, $C' = [0, 0, 1, 0, 0, 0, 0]$ and $D' = \frac{1}{3}(A' + B' + C')$. For ease of notation, we define a mapping $\mathrm{det} : \{A', B', C', D'\} \rightarrow T^{\mathrm{det}}$, with $\mathrm{det}(A') = A, \mathrm{det}(B') = B, \mathrm{det}(C') = C$ and $\mathrm{det}(D') = D$. We also introduce four new types, $E' = [0, 0, 0, 1, 0, 0, 0]$, $F' = [0, 0, 0, 0, 1, 0, 0]$, $G' = [0, 0, 0, 0, 0, 1, 0]$ and $H' = [0, 0, 0, 0, 0, 0, 1]$. Define $T^{\mathrm{rand}} = \{A', B', C', D', E', F', G'\}$, and $\boldsymbol{T}^{\mathrm{rand}} = T^{\mathrm{rand}} \times T^{\mathrm{rand}}$.

We now define x^{rand}, which is again symmetric, in the sense that $x_1^{\mathrm{rand}}(V_1, V_2) = x_2^{\mathrm{rand}}(V_2, V_1)$ for every $V_1, V_2 \in T^{\mathrm{rand}}$. We therefore only describe x_1^{rand}. When both agents report types in $\{A', B', C', D'\}$, the first three coordinates of each agent's allocation are given by x^{det} when fed by the corresponding types in T^{det}, and the remaining coordinates are filled in according to the opponent's report. More specifically,

$$\forall V_1 \in \{A', B', C', D'\},$$
$$x_1^{\mathrm{rand}}(V_1, A') = [x_1^{\mathrm{det}}(\mathrm{det}(V_1), A), 0, 100, 100, 100],$$
$$x_1^{\mathrm{rand}}(V_1, B') = [x_1^{\mathrm{det}}(\mathrm{det}(V_1), B), 100, 0, 100, 100],$$
$$x_1^{\mathrm{rand}}(V_1, C') = [x_1^{\mathrm{det}}(\mathrm{det}(V_1), C), 100, 100, 0, 100],$$
$$x_1^{\mathrm{rand}}(V_1, D') = [x_1^{\mathrm{det}}(\mathrm{det}(V_1), D), 100, 100, 100, 0] \quad (4)$$

For the other types, we have

$$\forall V_1 \in \{E', F', G', H'\}, \forall V_2 \in \{A', B', C', D'\},$$
$$x_1^{\mathrm{rand}}(V_1, V_2) = x_1^{\mathrm{rand}}(C', V_2).$$
$$\forall V_2 \in \{E', F', G', H'\}, \forall V_1 \in T^{\mathrm{rand}},$$
$$x_1^{\mathrm{rand}}(V_1, V_2) = [0, 0, 0, 100, 100, 100, 100].$$

Note that $\boldsymbol{x}^{\mathrm{rand}}$ itself is deterministic. The difficulty we need to overcome in this section is that the *extension* of $\boldsymbol{x}^{\mathrm{rand}}$ may be randomized, and we must show that any extension to $\mathrm{Conv}(\boldsymbol{T}^{\mathrm{rand}})$ cannot be DSIC.

The set of feasible allocations is $\mathscr{F} = \{\boldsymbol{x}^{\mathrm{rand}}(V_1, V_2)\}_{V_1, V_2 \in T^{\mathrm{rand}}}$. Again we emphasize that the set of feasible allocations is *not* a product space.

We first show that a subset of the payments are still "locked" as they were in the deterministic setting.

Lemma 4. *For any allocation rule $\tilde{x}$ that is an extension of $\boldsymbol{x}^{\mathrm{rand}}$, if $\tilde{x}$ can be implemented by a DSIC mechanism with payment rule $\boldsymbol{p}$, then for any $V \in \{A', B', C', D'\}$ and any $V', V'' \in T^{\mathrm{rand}}$, $p_1(V', V) = p_1(V'', V)$, and $p_2(V, V') = p_2(V, V'')$.*

Proof. The proof for types in $\{A', B', C', D'\}$ follows the same steps as Lemma 1. For any type in $\{E', F', G', H'\}$ note that they receive the same allocation as type C' and therefore if they are charged a payment that is different from the payment C' is charged than either C' or that type would be incentivized to deviate. $\square$

In order to have a DSIC convex extension in this setting we must satisfy a condition similar to lemma 2 with the higher dimensional versions of the types from $\boldsymbol{T}^{\mathrm{det}}$.

Lemma 5. *If $\tilde{x}$ is a DSIC extension of $\boldsymbol{x}^{\mathrm{rand}}$, then for $t_1 = \frac{1}{3}A' + \frac{1}{3}B' + \frac{1}{3}D'$, and any $V_2 \in \{A', B', C', D'\}$, $\tilde{x}_1(t_1, V_2)$ is supported on $\{\tilde{x}_1(A', V_2), \tilde{x}_1(B', V_2), \tilde{x}_1(C', V_2), \tilde{x}_1(D', V_2)\}$.*

Proof. For the sake of contradiction, assume $\tilde{x}$ is a DSIC extension of $\boldsymbol{x}^{\mathrm{rand}}$, implementable by payment rule $\boldsymbol{p}$, and for some $V_2 \in \{A', B', C', D'\}$,

there exist a pair of types V_1, V_2' such that $\mathbf{Pr}[\tilde{\boldsymbol{x}}(t_1, V_2) = \boldsymbol{x}^{\mathrm{rand}}(V_1, V_2')] > 0$ for a $V_2' \neq V_2$. Let X_{V_2} denote the set of allocations $\{\tilde{x}_1(A', V_2), \tilde{x}_1(B', V_2), \tilde{x}_1(C', V_2), \tilde{x}_1(D', V_2)\}$.

Let $\alpha = \mathbf{Pr}[\tilde{x}_1(t_1, V_2) \in X_{V_2}]$ and $\beta = \mathbf{Pr}[\tilde{x}_1(t_1, V_2) \notin X_{V_2}]$. Then by assumption, $\beta > 0$.

Also let $\tilde{x}_1^\alpha(t_1) = \mathbf{E}[\tilde{x}_1(t_1, V_2) \mid \tilde{x}_1(t_1, V_2) \in X_{V_2}]$ and $\tilde{x}_1^\beta(t_1) = \mathbf{E}[\tilde{x}_1(t_1, V_2) \mid \tilde{x}_1(t_1, V_2) \notin X_{V_2}]$. Then the expected value for agent 1 truthfully reporting t_1 is $\langle t_1, \alpha\tilde{x}_1^\alpha(t_1) + \beta\tilde{x}_1^\beta(t_1)\rangle$.

In order for $\tilde{x}$ to be DSIC, it must provide type t_1 with at least as much utility as it could receive from deviating to any type in T^{rand}.

Let $V_{\max} \in \mathrm{argmax}_{V_1' \in T^{\mathrm{rand}}} \langle t_1, x_1^{\mathrm{rand}}(V_1', V_2))\rangle$ then $\tilde{x}$ must satisfy the following DSIC constraint,

$$\alpha\langle t_1, \tilde{x}_1^\alpha(t_1)\rangle + \beta\langle t_1, \tilde{x}_1^\beta(t_1)\rangle - p_1(t_1, V_2) \geq$$
$$\langle t_1, x_1^{\mathrm{rand}}(V_{\max}, V_2))\rangle - p_1(V_{\max}, V_2). \quad (5)$$

We use this constraint to place an upper bound on $p_1(t_1, V_2)$.

By definition of $V_{\max}$,

$$\langle t_1, x_1^{\mathrm{rand}}(V_{\max}, V_2))\rangle \geq \langle t_1, \tilde{x}_1^\alpha(t_1)\rangle.$$

Subtracting $\langle t_1, x_1^{\mathrm{rand}}(V_{\max}, V_2))\rangle$ from both sides of eq. (5) and utilizing the fact that $\alpha + \beta = 1$ we have

$$\beta\langle t_1, \tilde{x}_1^\beta(t_1)\rangle - \beta\langle t_1, x_1^{\mathrm{rand}}(V_{\max}, V_2)\rangle$$
$$\geq p_1(t_1, V_2) - p_1(V_{\max}, V_2).$$

Now using the fact that t_1's maximum difference in value for any two allocations in $\mathscr{F}$ is upper bounded by 3, we get

$$3\beta + p_1(V_{\max}, V_2) \geq p_1(t_1, V_2). \quad (6)$$

By construction, for $V_2 \in \{A', B', C', D'\}$, there exists a type $V_{\mathrm{not}} \in \{E', F', G', H'\}$ which has value 0 for every allocation in X_{V_2}. (For example if $V_2 = A'$ then all of the allocations in $\{\tilde{x}_1(A', A'), \tilde{x}_1(B', A'), \tilde{x}_1(C', A'), \tilde{x}_1(D', A')\}$ have a 0 in the 4$^{\mathrm{th}}$ coordinate and in this case $V_{\mathrm{not}} = E'$.) Now by Lemma 4, V_{not} receives utility $-p_1(V_{\mathrm{not}}, V_2) = -p_1(V_{\max}, V_2)$ when reporting truthfully against opponent type V_2.

Therefore, in order to keep V_{not} from deviating to type t_1, we must satisfy the following DSIC constraint,

$$-p_1(V_{max}, V_2) \geq \alpha\langle V_{\mathrm{not}}, \tilde{x}_1^\alpha(t_1)\rangle + \beta\langle V_{\mathrm{not}}, \tilde{x}_1^\beta(t_1)\rangle$$
$$- p_1(t_1, V_2). \quad (7)$$

Since $\tilde{x}_1^\alpha(t_1)$ is the expected allocation conditioned on the resulting allocation being an element of X_{V_2}, and V_{not} has zero value for any allocation in X_{V_2}, we have $\langle V_{\mathrm{not}}, \tilde{x}_1^\alpha\rangle = 0$.

Therefore,

$$p_1(t_1, V_2) \geq \beta\langle V_{\mathrm{not}}, \tilde{x}_1^\beta(t_1)\rangle + p_1(V_{\max}, V_2). \quad (8)$$

Notice that for every allocation outside of X_{V_2} the type V_{not} has value exactly 100 and therefore $\langle V_{\mathrm{not}}, x_\beta(t)\rangle = 100$. We now use this fact along with eq. (6) and eq. (8) and derive the following contradiction,

$$3\beta + p_1(V_{\max}, V_2) - p_1(V_{\max}, V_2) \geq \beta\langle V_{\mathrm{not}}, x_\beta\rangle,$$
$$\Rightarrow \quad 3\beta \geq 100\beta.$$

Since by assumption $\beta > 0$, this is a contradiction.

$\square$

We are now ready to prove Theorem 3.

Proof of Theorem 3. Consider $t_1 = \frac{1}{3}A' + \frac{1}{3}B' + \frac{1}{3}D'$ and assume $\mathbf{Pr}[\tilde{x}_1(t_1, A') = x_1^{\mathrm{rand}}(A', A')] < 1$.

We know from Lemma 5 that $\tilde{x}_1(t_1, A')$ is supported on $\{\tilde{x}_1(A', A'), \tilde{x}_1(B', A'), \tilde{x}_1(C', A'), \tilde{x}_1(D', A')\}$. Since $x_1^{\mathrm{rand}}(A', A')$ is the only allocation for which both A' and B' have positive value, t_1 satisfies

$$\forall V' \neq A', \quad \langle t_1, x_1^{\mathrm{rand}}(V', A')\rangle < \langle t_1, x_1^{\mathrm{rand}}(A', A')\rangle. \quad (9)$$

But in order for t_1 to have no incentive to report A' when agent 2's type is A', we have

$$\mathbf{E}\left[\langle t_1, \tilde{x}_1(t_1, A')\rangle\right] - p_1(t_1, A') \geq$$
$$\langle t_1, x_1^{\mathrm{rand}}(A', A')\rangle - p_1(A', A').$$

Using eq. (9), we have

$$p_1(A', A') > p_1(t_1, A'). \qquad (10)$$

We now show that, similar to Lemma 3, a strictly lower payment for t_1 as in Equation (10) would violate DSIC constraints. The DSIC constraint that keeps type D' from deviating to type t_1 when the opponents type is A' is

$$\langle D', x_1^{\text{rand}}(D', A') \rangle - p_1(D', A')$$
$$\geq \langle D', \tilde{x}_1(t_1, A') \rangle - p_1(t_1, A'). \qquad (11)$$

Since $\tilde{x}_1(t_1, A')$ is supported on $\{\tilde{x}_1(A', A'), \tilde{x}_1(B', A'), \tilde{x}_1(C', A'), \tilde{x}_1(D', A')\}$ and type D' values all of these allocations equally we have

$$\langle D', \tilde{x}_1(t_1, A') \rangle = \langle D', x_1^{\text{rand}}(D', A') \rangle,$$

and by Lemma 4 we have $p_1(D', A') = p_1(A', A')$. Using these facts we write Equation (11) as

$$p_1(t_1, A') \geq p_1(A', A').$$

This contradicts Equation (10).

Therefore, $\tilde{x}_1(t_1, A')$ must be deterministically $x_1^{\text{rand}}(A', A')$. But the only allocation in $\mathscr{F}$ with agent 1's allocation being this is $\boldsymbol{x}^{\text{rand}}(A', A')$, and hence $\tilde{x}_2(t_1, A')$ is deterministically $[1, 1, 0, 0, 100, 100, 100]$. Using the same argument, one can show $\tilde{x}_1(t_1, D') = [4, 4, 0, 100, 100, 100, 0]$ with probability 1, which implies $\tilde{x}_2(t_1, D')$ is supported on $\{[2, 2, 0, 100, 0, 100, 100], [4, 4, 0, 100, 100, 100, 0]\}$. For any allocation with this support, weak monotonicity is violated for agent 2's types A' and D'. □

Revenue Gap

In this section we explore the revenue gap between mechanisms that satisfy DSIC and IR on $\text{supp}(D)$ and mechanisms that satisfy DSIC and IR on $\text{Conv}(\text{supp}(D))$. Any type that is in $\text{Conv}(\text{supp}(D))$ and not $\text{supp}(D)$ occurs with probability zero and does not itself contribute to the revenue. It is therefore not obvious that enforcing constraints on these additional types will impact revenue. In fact, Theorem 1 shows that if the feasible set is downward closed there is no gap in revenue between these two typespaces. However, for arbitrary feasibility sets, a gap can arise because not every mechanism which is DSIC and IR can be extended to $\text{Conv}(\text{supp}(D))$ without changing the payment rules on the support. We show that for some feasibility sets and distributions the gap in revenue between the optimal DSIC and IR mechanism on $\text{supp}(D)$ and the optimal mechanism on $\text{Conv}(\text{supp}(D))$ is unbounded.

For a feasible set of allocations $\mathscr{F}$ and a distribution $\mathbf{D}$ we, let $\text{OPT}(\mathscr{F}, \mathbf{D})$ denote the optimal revenue extractable by a DSIC and IR mechanism on $\text{supp}(D)$, and let $\widetilde{\text{OPT}}(\mathscr{F}, \mathbf{D})$ denote the maximum revenue extractable by a mechanism that is DSIC and IR on $\text{Conv}(\text{supp}(D))$.

We focus on the single agent case and show that the ratio between $\text{OPT}(\mathscr{F}, D)$ and $\widetilde{\text{OPT}}(\mathscr{F}, D)$ is unbounded.

Theorem 5. $\forall \alpha$, there exists a feasible set $\mathscr{F}$ and distribution $\mathbf{D}$ such that $\text{OPT}(\mathscr{F}, \mathbf{D}) \geq \alpha \widetilde{\text{OPT}}(\mathscr{F}, \mathbf{D})$.

Proof. Let $\vec{\mathbf{1}}$ be the "all-ones" vector of length k and define the set

$$\mathcal{S}_1 = \{t \in \mathbb{R}^k \mid \exists j \text{ s. t. } t_j = 1$$
$$\text{and } \forall h \neq j, \, t_h = 0\}.$$

Intuitively, $\mathcal{S}_1$ is the set of all vectors of length k with exactly one coordinate being 1 and the rest of the coordinates being zero. We define the support, $\text{supp}(D) = \mathcal{S}_1 \cup \vec{\mathbf{1}}$ and the set of feasible allocations $\mathscr{F} = \mathcal{S}_1$.

We can show that $\text{OPT}(\mathscr{F}, \mathbf{D}) = 1$ for any distribution $\mathbf{D}$ that shares this support. This is obtained by giving each $t \in \mathcal{S}_1$ an allocation x equal to their type i.e. $t = x(t)$ and giving type $t = \vec{\mathbf{1}}$ any allocation in the feasible set. We can now charge a payment of 1 to every type.

Without loss of generality assume $\alpha > 1$. Now for any α define $\epsilon < \frac{1}{\alpha}$ and consider the distribution where $Pr[t = \vec{\mathbf{1}}] = 1 - \epsilon$ and $Pr[t \neq \vec{\mathbf{1}}] = \epsilon$. One of the types contained in $\text{Conv}(\text{supp}(D))$ is $t^k = [\frac{1}{k}, \frac{1}{k} \ldots \frac{1}{k}]$. t^k has a value of $\frac{1}{k}$ for every feasible allocation and every lottery of existing allocations, therefore the maximum payment we can charge without violating IR constraints is $\frac{1}{k}$. However, the all ones type has value 1 for every feasible allocation and can now gain utility of $\frac{k-1}{k}$ by deviating to type t^k. Therefore to maintain incentive compatibility we must lower the price we charge the all ones type to $\frac{1}{k}$ (as well as the payment charged to one of the types $\mathcal{S}_1$). Therefore the revenue generated on the convex hull if we take k to be large is, $\widetilde{\text{OPT}}(\mathscr{F}, \mathbf{D}) \leq \lim_{k \to \infty} (1 - \epsilon)(\frac{1}{k}) + \epsilon \leq \frac{1}{\alpha}$. □

Conclusion and Future Work

We presented a DSIC mechanism on a discrete type space which cannot be implemented if DSIC is required on the convex hull of its type space. Extension from discrete to convex domains is a common step in automated mechanism design, and it is unfortunate that, as an implication of our result, there exists no general procedure for extending an arbitrary DSIC mechanism to the convex hull of its domain. However, as we showed, such extensions are possible in specific settings, with special type spaces (e.g. single dimensional) or special feasible set of allocations (e.g. downward closed or, more generally, single swap feasible). Other sufficient conditions for extensibility should be useful to have.

Another possibility we leave open is conditions on an allocation rule (rather than the type space or the set of feasible allocations) that would guarantee its extensibility. We know that an allocation rule is DSIC implementable if and only if it is cyclic monotone. Is there a condition stronger than cyclic monotonicity which guarantees both implementability and extensibility?

The mechanism we constructed for the impossibility result is intricate. A third question we leave open is whether one may find a succinct property that is *necessary* for extensibility; in other words, is there a natural condition whose violation must result in inextensible mechanisms?

References

Abebe, R.; Cole, R.; Gkatzelis, V.; and Hartline, J. D. 2019. A truthful cardinal mechanism for one-sided matching. *CoRR* abs/1903.07797.

Archer, A., and Kleinberg, R. 2014. Truthful germs are contagious: A local-to-global characterization of truthfulness. *Games and Economic Behavior* 86:340–366.

Ashlagi, I.; Dobzinski, S.; and Lavi, R. 2012. Optimal lower bounds for anonymous scheduling mechanisms. *Mathematics of Operations Research* 37(2):244–258.

Balcan, M.; Sandholm, T.; and Vitercik, E. 2018. A general theory of sample complexity for multi-item profit maximization. In *Proceedings of the 2018 ACM Conference on Economics and Computation,EC 2018 Ithaca, NY, USA, June 18-22, 2018*, 173–174.

Chawla, S.; Hartline, J. D.; Malec, D. L.; and Sivan, B. 2010. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, 311–320.

Chawla, S.; Hartline, J. D.; Malec, D. L.; and Sivan, B. 2013. Prior-independent mechanisms for scheduling. In *Symposium on Theory of Computing Conference, STOC 2013, Palo Alto, CA, USA, June 1-4, 2013*, 51–60.

Chawla, S.; Malec, D. L.; and Sivan, B. 2010. The power of randomness in bayesian optimal mechanism design. In *Proceedings 11th ACM Conference on Electronic Commerce, EC 2010,Cambridge, Massachusetts, USA, June 7-11, 2010*, 149–158.

Clarke, E. 1971. Multipart pricing of public goods. *Public Choice* 11(1):17–33.

Conitzer, V., and Sandholm, T. 2002. Complexity of mechanism design. In *Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence,UAI 2002, University of Alberta, Edmonton, Alberta, Canada, August 1-4, 2002*, 103–110.

Daskalakis, C.; Deckelbaum, A.; and Tzamos, C. 2017. Strong duality for a multiple-good monopolist. *Econometrica* 85(3):735–767.

Devanur, N. R.; Mihail, M.; and Vazirani, V. V. 2005. Strategyproof cost-sharing mechanisms for set cover and facility location games. *Decision Support Systems* 39(1):11–22.

Dobzinski, S.; Fu, H.; and Kleinberg, R. 2015. Approximately optimal auctions for correlated bidders. *Games and Economic Behavior* 92:349–369.

Feldman, M.; Gravin, N.; and Lucier, B. 2015. Combinatorial auctions via posted prices. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, 123–135.

Feng, Y., and Hartline, J. D. 2018. An end-to-end argument in mechanism design (prior-independent auctions for budgeted agents). In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, 404–415.

Groves, T. 1973. Incentives in teams. *Econometrica* 41(4):617–31.

Guo, M., and Conitzer, V. 2010. Computationally feasible automated mechanism design: General approach and case studies. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*.

Likhodedov, A., and Sandholm, T. 2004. Methods for boosting revenue in combinatorial auctions. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, AAAI 2004 July 25-29, 2004, San Jose, California, USA*, 232–237.

Likhodedov, A., and Sandholm, T. 2005. Approximating revenue-maximizing combinatorial auctions. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference,AAAI 2005, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, 267–274.

Morgenstern, J., and Roughgarden, T. 2016. Learning simple auctions. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, 1298–1318.

Myerson, R. B. 1981. Optimal auction design. *Mathematics of Operations Research* 6(1):58–73.

Nisan, N., and Ronen, A. 2001. Algorithmic mechanism design. *Games and Economic behavior* 35(1-2):166–196.

Rochet, J.-C. 1987. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics* 16(2):191 – 200.

Saks, M. E., and Yu, L. 2005. Weak monotonicity suffices for truthfulness on convex domains. In *Proceedings 6th ACM Conference on Electronic Commerce, EC 2005, Vancouver, BC, Canada, June 5-8, 2005*, 286–293.

Sandholm, T., and Likhodedov, A. 2015. Automated design of revenue-maximizing combinatorial auctions. *Operations Research* 63(5):1000–1025.

Vickrey, W. 1961. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance* 16(1):8–37.