Partial Verification as a Substitute for Money

Sofia Ceppi,[1] **Ian Kash,**[2] **Rafael Frongillo**[3]
[1]PROWLER.io, [2]University of Illinois at Chicago
[3]University of Colorado Boulder
sofia@prowler.io, iankash@uic.edu, raf@colorado.edu

Abstract

Recent work shows that we can use partial verification instead of money to implement truthful mechanisms. In this paper we develop tools to answer the following question. Given an allocation rule that can be made truthful with payments, what is the minimal verification needed to make it truthful without them? Our techniques leverage the geometric relationship between the type space and the set of possible allocations.

Introduction

Mechanism design studies how to realize desirable outcomes to optimization problems in settings with self-interested agents. The most common tool to achieve desirable outcomes is the use of payments, and there is a large literature focusing on the following question. Given an allocation rule, which specifies the outcome that should be selected given the types of the agents, do there exist payments to turn it into an incentive compatible mechanism (to *implement* it) (Guesnerie and Laffont 1984; Saks and Yu 2005; Ashlagi et al. 2010; Frongillo and Kash 2014)?

Recent work has identified partial verification as a useful alternative to money to implement incentive compatible mechanisms. The idea is that the mechanism designer can detect some possible agent misreports, either by preventing them outright, or by penalizing the agent (e.g., by excluding her from the market). The power of partial verification is that the mechanism designer need not provide agents with incentives for a subset of the possible types they can report if verification of these types is in place. An example of such a partial verification is agents not being able to report a higher valuation for any assignment than is true. They are free, however, to report a lower value.

This specific type of verification has been adopted by Fotakis, Krysta and Ventre (2014). They consider the case in which a government is auctioning business licenses for cities under its administration, and companies want to get a license for some subset of cities to sell their stock of goods to the market. The verification assumption is that the government, that acts as auctioneer, can verify if the winner actually has sufficient goods in stock. In a particularly suggestive result, they showed that this verification suffices to implement all allocation rules that are implementable with money in single-minded combinatorial auctions. (In fact they show that a weaker verification, where the agent cannot overbid only on the bundle received, suffices.) Interestingly, this verification no longer suffices for agents who are k-minded, $k \geq 2$. Other work in the literature (see Related Work) has focused on specific scenarios like facility location and combinatorial auctions, and identifies the sets of verification assumptions that guarantee incentive compatibility of mechanisms.

Our work takes a different approach, in that we build tools to understand the power of verification independent of a specific scenario. In particular, we answer the following question: given an allocation rule that can be made truthful with payments, what is the minimal verification needed to make it truthful without them? Essentially, similar to (Ferraioli, Serafino, and Ventre 2016), we aim to inform the designer of the resources needed for verification. In contrast to this work, which focuses on facility location, we propose a geometric characterization of the verification needed to use any implementable-with-payment allocation rule in a scenario without transfers, while guaranteeing strategyproofness.

We introduce the concept of the *harmless set of types* as those which do not need to be verified for a given set of single-agent allocation rules. Our basic building block is a characterization of the structure of harmless sets for allocation rules which only assign two possible allocations and are implementable with payments. We then show how this can be extended to characterize harmless sets for more general sets of single-agent implementable-with-payments allocation rules. Our focus on sets of rules derives from the observation that multiagent allocation rules are, from the perspective of a single agent, just a set of allocation rules parameterized by the types of the other agents.

Our characterization highlights a split in the nature of harmless sets of types for deterministic and randomized mechanisms. Deterministic and universally truthful mechanisms both have large harmless sets of types, while in contrast the harmless set for truthful-in-expectation mechanisms is quite restricted. Our results are constructive and provide geometric insights for these findings.

The central contribution of our approach is its generality: our analysis could in principle be applied to any mechanism or class of mechanisms. Moreover, while our results are often stated for two allocations and single-agent settings, they also apply directly to more than two allocations and multi-agent settings, by standard arguments. We also examine two extensions: allocation-dependent verification, a weaker form of verification that can expand the harmless set of types, and reverse verification, where the reported type of the agent is considered when computing the types that needs to be verified. We conclude with examples showing how our approach can be used in two application domains and how our results replicate and extend existing results in the literature. See the extended version of this paper on ArXiv [1] for further discussion about the extensions, more examples, and missing proofs.

Related Work

Several works in both the economics and computer science literatures focus on the design of incentive compatible mechanisms with verification (Green and Laffont 1986; Fotakis, Krysta, and Ventre 2015; Fotakis and Zampetakis 2015; Penna and Ventre 2009; Ventre 2014) to overcome the Gibbard-Satterthwaite impossibility result (Gibbard 1973; Satterthwaite 1975) for mechanisms without money. In particular, given a mechanism, these works aim to reduce the types agents can report to the ones that do not bring benefit to them. This is done by either assuming that the reportable type space varies depending on the true type of the agents or that the mechanism can verify part of the type space and penalize agents that misreport in that space.

The power of verification in the design of mechanisms without money has been studied in a number of applications including scheduling of unrelated machines (Koutsoupias 2014), combinatorial auctions (Krysta and Ventre 2015), and assignment and allocation problems (Dughmi and Ghosh 2010; Guo and Conitzer 2010). A particular focus has been on the design of mechanisms with verification yielding good approximate solutions to the problem of facility location on a line (Procaccia and Tennenholtz 2009; Serafino and Ventre 2016; 2014; Ferraioli, Serafino, and Ventre 2016). Much of this literature has focused on identifying verifications which seem natural for a particular application (Koutsoupias 2014; Serafino and Ventre 2016; Fotakis, Krysta, and Ventre 2014), in contrast to the present work which fixes a set of allocation rules and asks what verification would be minimally necessary to render the mechanisms truthful. Most similarly to our own work, Ferraioli et al. (2016) have considered the question the minimum set of assumptions needed in the facility location setting.

Preliminaries

In this section, we focus on mechanism design with a single agent. Let S denote the set of assignments, one of which the agent will receive, $|S| = m$. Let A denote the set of allocations, where an allocation $a \in A$ is a probability distribution over assignments. Formally, $A \subseteq \{a \in [0,1]^m : \sum_{s \in S} a(s) = 1\}$. One can think of assignments as a set of mutually exclusive outcomes, and allocations as distributions over these outcomes, which for example will be point distributions when we consider deterministic mechanisms.

We use $\theta = [\theta_{s_1}, \theta_{s_2}, ..., \theta_{s_m}] \in \mathbb{R}^m$ to denote the type of the agent, i.e., her private information, where θ_{s_i} is the agent's value for the assignment s_i. The set of possible agent types is denoted $\Theta \subseteq \mathbb{R}^m$. A single-agent direct revelation mechanism is denoted $\mathcal{M} = \{f, p\}$, where $f : \Theta \to A$ is the allocation rule and $p : \Theta \to \mathbb{R}$ is the payment rule. Under this mechanism, the utility of an agent with type θ who reports type $\hat{\theta}$ is $u^{\mathcal{M}}(\theta, \hat{\theta}) = f(\hat{\theta}) \cdot \theta - p(\hat{\theta})$.

We now introduce several terms and definitions.

Incentive Compatibility. A mechanism is incentive compatible (i.e., truthful) if the agent is incentivized to communicate to the mechanism her true type. Since agents are rational, to guarantee incentive compatibility, the mechanism must guarantee that each agent is better off when she reports her true type than when she misreports, i.e., $u^{\mathcal{M}}(\theta, \theta) \geq u^{\mathcal{M}}(\theta, \hat{\theta})$ for all $\theta, \hat{\theta} \in \Theta$.

Implementable-with-Payments Allocation Rule. An allocation rule f is implementable-with-payments if there exists a payment rule p such that the mechanism $\mathcal{M} = \{f, p\}$ is incentive compatible.

Deterministic Mechanism. A mechanism is deterministic if each allocation $a \in A$ selects one assignment with probability 1 and all the other outcomes with probability 0, i.e., $a(s) = 1, s \in S$ and $a(s') = 0, \forall s' \neq s, s' \in S, \forall a \in A$. In this sense, all mechanisms, including deterministic mechanisms, are randomized.

Universally Truthful Mechanism. We say an incentive compatible randomized mechanism $\mathcal{M} = (f, p)$ is universally truthful if it is a distribution over incentive compatible deterministic mechanisms. That is, there is a set of deterministic incentive compatible mechanisms $S_{\mathcal{M}}$ known as the support of $\mathcal{M}$ and a probability distribution α such that $f_{\mathcal{M}} = \sum_{\mathcal{M}' \in S_{\mathcal{M}}} \alpha_{\mathcal{M}'} f_{\mathcal{M}'}$.

Truthful in Expectation Mechanism. If a randomized mechanism is incentive compatible, we say that it is truthful in expectation. (The expected value is implicit in the dot product in the definition of $u^{\mathcal{M}}(\theta, \hat{\theta})$.)

Truthful with Verification Mechanism. Let $V \subseteq \Theta \times \Theta$ be the set of pairs $(\theta, \hat{\theta})$ that the designer can verify and denote with $\mathcal{M}_v = (f, V)$ a mechanism with verification. $\mathcal{M}_v$ is truthful if for all pairs of types $(\theta, \hat{\theta})$ either $f(\theta) \cdot \theta \geq f(\hat{\theta}) \cdot \theta$ or $(\theta, \hat{\theta}) \in V$. Intuitively, either an agent with type θ prefers not to report $\hat{\theta}$ or the mechanism designer can detect or prevent such a report.

Basics of Partial Verification

In this section, we develop our basic tools for reasoning about what types do not need to be verified. In particular, given a type θ, if the agent can never benefit by reporting some other type $\hat{\theta}$, then from the perspective of the mechanism designer it is unnecessary to be able to verify $(\theta, \hat{\theta})$,

[1] https://arxiv.org/abs/1812.07312

i.e., verify that $\hat{\theta}$ is not the agent's true type. We call such types *harmless*. [2]

Definition 1. *Given a type θ and allocation rule f, the* harmless set of types $H(\theta, f)$ *is the set composed by the types $\hat{\theta} \in \Theta$ such that $f(\theta) \cdot \theta \geq f(\hat{\theta}) \cdot \theta$.*

We can also talk about the harmless set of types for multiple allocation rules.

Definition 2. *Let F be a set of allocation rules. Then the* harmless set of types $H(\theta, F)$ *is the set composed by the types $\hat{\theta} \in \Theta$ such that $f(\theta) \cdot \theta \geq f(\hat{\theta}) \cdot \theta$ for every $f \in F$.*

It is immediate from this definition that the harmless set of types of a set of allocation rules is the intersection of their individual harmless sets. This is because our definition imposes a strong requirement for a type to be harmless: it identifies a type as harmless only if it is harmless for every allocation rule in the set. That is, there is no scenario under which the agent can benefit from reporting a harmless type. This strong definition is in the same spirit as the definition of incentive compatibility; we only want to declare a type harmless if the mechanism designer never has to worry about that type being reported.

Observation 1. *The harmless set of types $H(\theta, F)$ corresponds to the intersection of the harmless set of types of every allocation rule $f \in F$, i.e., $H(\theta, F) = \cap_{f \in F} H(\theta, f)$.*

Using $H(\theta, F)$, we can express the minimal verification needed to guarantee that a implementable-with-payments allocation rule is also truthful without them. In particular, this minimal verification corresponds to the set V where $(\theta, \hat{\theta}) \in V$ if and only if $\hat{\theta} \in \Theta \setminus H(\theta, f)$.

Harmless Sets with Two Allocations (Informally)

We now introduce our main tools for understanding the harmless sets of implementable-with-payments allocation rules. At first, we characterize the harmless sets of implementable-with-payments allocation rules which have exactly two allocations in their range, and then show that there is a sense in which this captures everything we need to know about the harmless set of types even when there are more than two allocations. Before giving a formal treatment of this setting, we walk through it more informally.

With only two allocations, incentive compatible mechanisms have a simple, well-known form. By the taxation principle, any incentive compatible mechanism consists of assigning a price to each allocation and letting the agent choose which allocation it prefers to pay for (Guesnerie and Laffont 1984). Thus, if we call the two allocations a_1 and a_2 and assign them prices $p_1 = p(a_1)$ and $p_2 = p(a_2)$, an agent with type θ can be assigned $f(\theta) = a_1$ by an incentive compatible mechanism only if $a_1 \cdot \theta - p_1 \geq a_2 \cdot \theta - p_2$ (and similarly for a_2). Rewriting, it is easy to see that the types that are indifferent and could be assigned either allocation are those who satisfy $(a_1 - a_2) \cdot \theta = p_1 - p_2$. That is, these types all lie on a hyperplane. Further, the two half spaces on either side of this hyperplane correspond to the sets of types that prefer each allocation at the given prices, i.e., if θ is in the interior of one halfspace and θ' is in the interior of the other then $u^{\mathcal{M}}(\theta, \theta) > u^{\mathcal{M}}(\theta, \theta')$ and $u^{\mathcal{M}}(\theta', \theta') > u^{\mathcal{M}}(\theta', \theta)$.

Since such a hyperplane is uniquely identified by a *relative price* $c = p_1 - p_2$, every implementable-with-payments allocation rule f can be associated with the hyperplane $(a_1 - a_2) \cdot \theta = c$ for some real number c. Note however, that there will in general be many allocation rules associated with a single hyperplane because types on the hyperplane are indifferent between the allocations whose prices difference is c and so can be assigned to either allocation by an implementable-with-payments allocation rule.

Now consider a particular such f and a type θ. There are five possible cases for $H(\theta, f)$.

Case 1: $\theta \cdot a_1 > \theta \cdot a_2$ and $f(\theta) = a_1$. An agent with type θ is already receiving her preferred outcome, so the agent cannot gain by reporting another type. Thus $H(\theta, f) = \Theta$.

Case 2: $\theta \cdot a_1 > \theta \cdot a_2$ and $f(\theta) = a_2$. An agent with type θ can benefit by reporting any type θ' such that $f(\theta') = a_1$, so $H(\theta, f) = \Theta - \{\theta' : f(\theta') = a_1\}$. By the above analysis, $H(\theta, f)$ contains all types on the side of the hyperplane associated with f where types prefer a_2 at relative price c implied by f. It may also contain some types on the hyperplane, if f happens to assign them a_2.

Case 3: $\theta \cdot a_1 = \theta \cdot a_2$. This case is degenerate and the agent with type θ is totally indifferent between the two allocations, so $H(\theta, f) = \Theta$ regardless of the f chosen.

Cases 4 and 5: symmetric to Cases 1 and 2.

So what does $H(\theta, F)$ look like where F is the set of all such f? By Observation 1, we need to take the intersection of the harmless sets. In the degenerate case 3, this yields $H(\theta, F) = \Theta$. Otherwise, all that matters is the f for which case 2 applies. That is we care about the f which correspond to hyperplanes with c such that $(a_1 - a_2) \cdot \theta \leq c$. The intersection of the harmless sets of all these hyperplanes is the set of θ' which are "below" all of them. This is entirely determined by the "lowest" such hyperplane, the one where $(a_1 - a_2) \cdot \theta = c$. Consider the following example.

Example 1. *Consider the case of a deterministic incentive compatible mechanism with two possible assignments, s_1 and s_2, and two allocations, a_1 and a_2, such that $a_1(s_1) = 1$ and $a_2(s_2) = 1$, i.e., allocation a_1 assigns s_1 to the agent with probability 1, while allocation a_2 assigns s_2 with probability 1. Furthermore, assume that the agent's type is $\theta = (\theta_{s_1}, \theta_{s_2})$ with $\theta_{s_1} < \theta_{s_2}$. This setting is illustrated in Figure 1 (a), where $\theta_1 = \theta_{s_1}$ and $\theta_2 = \theta_{s_2}$.*

The hyperplane of types $\theta' \in \Theta$ for which $\theta'_{s_1} = \theta'_{s_2}$ is the 45 degree line from the origin, and which we refer to as the indifference hyperplane. *Note that it corresponds to taking $c = 0$, and that changing c just translates this line while keeping it at 45 degrees. The translations of this line for which $(a_1 - a_2) \cdot \theta \geq c$, i.e., $\theta_{s_1} - \theta_{s_2} \geq c$, are the lines that in the figure would be above θ; the lowest of these is the one which passes through θ, which we refer to as the* critical allocation hyperplane. *The harmless set $H(\theta, F)$ is*

[2] To simplify the characterization, we focus on harmless types instead of types that must be verified. Given a type θ and allocation rule f, a pair $(\theta, \hat{\theta})$ is in V if $\hat{\theta} \in \Theta \setminus H(\theta, f)$. For brevity, we say that the set of types which must be verified is $\Theta \setminus H(\theta, f)$. That is, the mechanism will be truthful as long as the mechanism designer can verify that the agent does not have true type θ when she reported θ', for all $\theta' \in \Theta \setminus H(\theta, f)$.

the set of types below this critical allocation hyperplane. It corresponds to the types that prefer s_1 relative to s_2 more strongly that θ. That is, those θ' where $\theta_{s_1} - \theta_{s_2} < \theta'_{s_1} - \theta'_{s_2}$.

Formal Treatment of Two Allocations

We define the concepts introduced in the previous section and formally prove how to identify the harmless set of types of implementable-with-payments allocation rules.

Definition 3. *An allocation rule $f_{\{a_i,a_j\}}$ is* separating *if $f_{\{a_i,a_j\}} : \Theta \to \{a_i, a_j\} \subseteq A$ and there exists a hyperplane which separates the type space Θ in two open half-spaces $\Theta', \Theta'' \subseteq \Theta$ such that the closure of their union is Θ and if $\theta \in \Theta'$ then$f(\theta) = a_i$ while if $\theta \in \Theta''$ then $f(\theta) = a_j$. (For brevity, when the allocation pair $\{a_i, a_j\}$ is clear from context we suppress it and simply write f.)*[3]

Definition 4. *Let $\bar{F}_{\{a_i,a_j\}}$ denote the set of separating allocation rules $f_{\{a_i,a_j\}}$. Then let $\bar{F} = \cup_{\{a_i,a_j\} \subseteq A} \bar{F}_{\{a_i,a_j\}}$.*

Given a separating allocation rule, we are interested in the hyperplane it induces, in the following sense.

Definition 5. *The allocation hyperplane $l_{f,A'}$ over allocation set $A' = \{a_i, a_j\}$ is the hyperplane that separates the two half-spaces identified by the separating allocation rule $f \in \bar{F}_{\{a_i,a_j\}}$. In the remaining of the paper, we will say that $l_{f,A'}$ is* induced by *the allocation rule $f \in \bar{F}_{\{a_i,a_j\}}$.*

Of course, selecting two allocations and a hyperplane is not sufficient for an allocation rule to be implementable-with-payments. By the taxation principle, the hyperplane must consist of all the types which are indifferent between the two allocations at a particular price. Further, the remaining types must receive the "correct" allocation. That is, those which would be willing to pay more than the price to get one allocation instead of the other are the ones that receive it. Such hyperplanes are exactly those parallel to the hyperplane of types indifferent between the two allocations.

Definition 6. *Given $\{a_i, a_j\} \subseteq A$, the* indifference hyperplane *$I_{\{a_i,a_j\}}$ is the hyperplane composed of types where the agent is indifferent between allocation a_i and allocation a_j, i.e. all the points $\theta' \in \Theta$ where $a_i \cdot \theta' = a_j \cdot \theta'$.*

Definition 7. *Let $L_{\{a_i,a_j\}}$ be the set of allocation hyperplanes $l_{f,\{a_i,a_j\}}$ parallel to indifference hyperplane $I_{\{a_i,a_j\}}$.*

The following observation formally summarizes the preceding discussion by showing that the hyperplanes in the set $L_{\{a_i,a_j\}}$ are only the ones that are induced by an implementable-with-payments allocation rules given the allocations $\{a_i, a_j\}$, and thus that the implementable-with-payments allocation rules are separating allocation rules.

Observation 2. *A hyperplane is in $L_{\{a_i,a_j\}}$ if and only if it is induced by an implementable-with-payments allocation rule $f \in \bar{F}_{\{a_i,a_j\}}$.*

While there are many allocation hyperplanes in $L_{\{a_i,a_j\}}$, the harmless set is entirely determined by one of them, the one identified as the "lowest" in the previous section, which we term the *critical allocation hyperplane.*

[3]Note that any implementable-with-payment allocation rule is also a separating allocation rule.

Definition 8. *The critical allocation hyperplane $l_{f^\theta,\{a_i,a_j\}} \in L_{\{a_i,a_j\}}$ is parallel to the indifference hyperplane $I_{\{a_i,a_j\}}$ and the agent's type belongs to it (i.e., $\theta \in l_{f^\theta,\{a_i,a_j\}}$). We call a rule $f^\theta \in \bar{F}_{\{a_i,a_j\}}$ that induces $l_{f^\theta,\{a_i,a_j\}}$ a critical allocation rule.*

Note that there exist an infinite number of critical allocation rules that induce a critical allocation hyperplane. In the remaining of the paper, the critical allocation rule we consider is the following: if $\theta' \in l_{f^\theta,\{a_i,a_j\}}$ and $a_i \cdot \theta' > a_j \cdot \theta'$, then $f^\theta(\theta') = a_i$ but $f^\theta(\theta) = a_j$. I.e., θ gets its less preferred allocation, while all the other types on the critical allocation hyperplane get the more preferred allocation. This implies that $\theta' \notin H(\theta, \bar{F}), \forall \theta' \neq \theta \in l_{f^\theta,\{a_i,a_j\}}$.

Lemma 1. *$H(\theta, \bar{F}_{\{a_i,a_j\}}) = H(\theta, f^\theta)$, i.e., the harmless set of types of the set of rules $\bar{F}_{\{a_i,a_j\}}$ is equal to the harmless set of types of allocation rule $f^\theta \in \bar{F}_{\{a_i,a_j\}}$ that induces the critical allocation hyperplane $l_{f^\theta,\{a_i,a_j\}}$.*

Proof. First notice that, since the allocation hyperplanes in $L_{\{a_i,a_j\}}$ are parallel, the critical allocation hyperplane $l_{f^\theta,\{a_i,a_j\}}$ divides the allocation hyperplanes in $L_{\{a_i,a_j\}}$ in two sets, depending on which side of it they lie. For those on the same side as the indifference hyperplane, the agent is already getting its preferred type so $H(\theta, f) = \Theta$. For those on the opposite side, the agent would rather report a type yielding her preferred allocation, so $H(\theta, f)$ corresponds to the open half-space containing the indifference hyperplane. The intersection of all these sets is exactly $H(\theta, f^\theta)$ □

The previous lemma implies that, given θ and a set of allocations $A' = \{a_i, a_j\}$, the critical allocation hyperplane $l_{f^\theta,A'}$ divides the space into two half-spaces, and that if $l_{f^\theta,A'} \neq I_{A'}$, then the open half-space containing the indifference hyperplane corresponds to the harmless set of types of $\bar{F}_{\{a_i,a_j\}}$, otherwise, if $l_{f^\theta,A'} = I_{A'}$, then $H(\theta, \bar{F}_{\{a_i,a_j\}}) = H(\theta, f^\theta) = \Theta$.

Definition 9. *Let $\bar{F}^\theta = \{f^\theta_{\{a_i,a_j\}} : \{a_i, a_j\} \in A\}$ be the set of critical allocation rules given all pairs $\{a_i, a_j\} \in A$.*

Definition 10. *Let $L^\theta = \{l_{f^\theta,\{a_i,a_j\}} : \{a_i, a_j\} \in A\}$ denote the set of critical allocation hyperplanes.*

From Observation 1 and Lemma 1, follows Corollary 1, which says that to identify the harmless set it suffices to identify the critical allocation hyperplanes.

Corollary 1. $H(\theta, \bar{F}) = H(\theta, \bar{F}^\theta) = \cap_{f^\theta \in \bar{F}^\theta} H(\theta, f^\theta)$.

Figure 1(a) shows an example of a critical allocation hyperplane, indifference hyperplane, and harmless set of types for a set of implementable-with-payments allocation rules for the case with two allocations.

This example also provides a clear geometric explanation for the phenomenon observed in previous work that "symmetric" verifications (which tend to take the form of a constraint that misreports must be local to the true type) do not tend to help while "asymmetric" ones do (Fotakis and Zampetakis 2015). Because θ is on the critical allocation hyperplane, there are arbitrarily close misreports which can lead

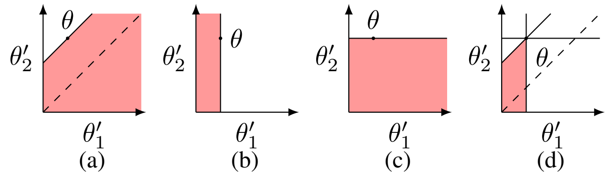

Figure 1: Harmless sets for individual pairs of allocations for types with $\theta'_\emptyset = 0$ and their intersection (right graph).

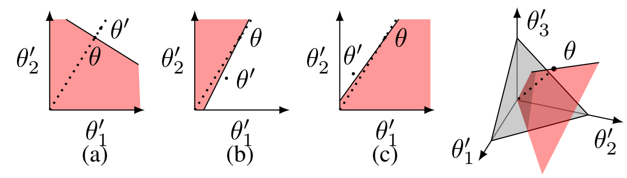

Figure 2: Harmless sets for randomized allocation rules. The dotted line depicts the scaled versions of θ, $\{\lambda\theta : \lambda \in \mathbb{R}\}$.

to a benefit for some allocation rules, so restricting misreports to be close to the true type does not help. In contrast, an asymmetric verification which rules out the entire half space above the critical allocation hyperplane is very useful.

More than Two Allocations

Now that we understand how to identify harmless sets of types of implementable-with-payments allocation rules with two allocations, we can extend our analysis to cases with more than two allocations. The key observation is that if a type θ' is not harmless then there exists an allocation rule f and choice of a_1 and a_2 such that $f(\theta') = a_1$ while $f(\theta) = a_2$ but $\theta \cdot a_1 > \theta \cdot a_2$. Since only these two allocations are relevant, we can actually find an implementable-with-payments allocation rule for which θ' is not harmless that only allocates a_1 and a_2, thus reducing to the case we have already analyzed. To identify the harmless set when there are more than two allocations, we thus intersect the harmless sets resulting from each pair of allocations.

Theorem 1. *Let F be the set of implementable-with-payments allocation rules. $H(\theta, F) = H(\theta, \bar{F}^\theta)$.*

Proof. Because $\bar{F}^\theta \subset F$, $H(\theta, F) \subset H(\theta, \bar{F}^\theta)$. For the other direction, let θ' be given such that $\theta' \notin H(\theta, F)$. By Definition 1, $\theta' \notin H(\theta, F)$ if and only if there exists an allocation rule $f \in F$ such that $\theta \cdot f(\theta') > \theta \cdot f(\theta)$. By the taxation principle, we can represent f by a list of allocations and the price for each allocation. Construct f' from f by eliminating all allocations except $f(\theta)$ and $f(\theta')$ from this list. Then f' is implementable-with-payments (those from the list), $f'(\theta) = f(\theta)$, $f'(\theta') = f(\theta')$, and $f' \in \bar{F}$. Thus $\theta' \notin H(\theta, \bar{F})$. By Corollary 1, $\theta' \notin H(\theta, \bar{F}^\theta)$. □

Verification and Randomization

In this section, we examine the implications of allowing randomization for implementing mechanisms using partial verification. We show that there is a significant harmless set shared by all deterministic mechanisms. Since universally truthful mechanisms are just distributions over deterministic mechanisms, this turns out to be true for them as well. However, the harmless set shared by all truthful in expectation mechanisms is quite limited.

Deterministic Mechanisms

We now study how to identify the harmless set of types for all truthful deterministic mechanisms. The result naturally follows from Theorem 1 as the intersection of the harmless sets of all deterministic mechanisms with two allocations.

Theorem 2. *Let F be the set of deterministic implementable-with-payments allocation rules. Then, $H(\theta, F)$ is the intersection of the half-spaces generated by all $l_{f^\theta,\{a_i,a_j\}} \in L^\theta$ containing the origin.*

We illustrate the theorem with the following example.

Example 2. *Consider a case with three assignments, one of which is null with no value. Thus, we have assignments $S = \{\emptyset, 1, 2\}$ and allocations $A = \{a_1, a_2, a_3\}$. Without loss of generality, assume that $a_1(\emptyset) = 1$, $a_2(1) = 1$, and $a_3(2) = 1$. Consequently, $\bar{F}^\theta = \{f^\theta_{\{a_1,a_2\}}, f^\theta_{\{a_1,a_3\}}, f^\theta_{\{a_2,a_3\}}\}$. The harmless sets for these allocation rules are shown in Figures 1 (b), (c), and (a) respectively. Since $H(\theta, \bar{F}^\theta)$, given by their intersection, is shown in Figure 1(d).*

This example also illustrates a key point about implementing the verifications found by our method: despite the infinite type space and infinite set of allocation rules, we can express the properties we need to verify in terms of a finite number of halfspace constraints, which gives reason to believe they may be verifiable in practical situations.

Universally Truthful Mechanisms

As previously discussed, the harmless set for all universally truthful mechanisms is the same as for all deterministic mechanisms. We observe that every universally truthful mechanism can be represented as a distribution over truthful deterministic mechanisms, and every deterministic mechanism is a universally truthful mechanism that chooses the specific deterministic mechanism with probability 1.

Theorem 3. *The harmless set of all single agent universally truthful mechanisms is equal to the harmless set of all single agent truthful deterministic mechanisms.*

Truthful in Expectation Mechanisms

Our results for deterministic and universally truthful mechanisms are relatively positive, in that there is a significant harmless set of types which do not require verification. For truthful in expectation mechanisms however, our results are much more negative. Essentially, the only types in the harmless set are those which are a scaling or translation by adding the same constant to the value of each assignment of the original type, except in the special case where the agent is indifferent among two or more assignments, which adds additional dimensions to the harmless set. For brevity, we state the theorem for the case where no such indifferences exist.

Theorem 4. *Let θ be such that $\theta_{s_i} \neq \theta_{s_j}$ for all i and j and $m \geq 3$. The harmless set of types of all single agent truthful in expectation mechanisms is $\{\lambda\theta + \lambda'\mathbb{1} : \lambda \leq 1, \lambda' \in \mathbb{R}\}$.*

The proof follows from a technical lemma that encompasses the case with indifferences as well as more general scenarios, and is stated and proved in the extended version of the paper. A direct intuition is shown in Figure 2. Parts (b) and (c) show that we can find pairs of allocations where the critical allocation hyperplane is arbitrarily close to the line between θ and the origin. So, by considering the intersection of the harmless types of all the possible pair of allocations, the resulting harmless types must all be along this line. Part (a) shows that types along the line from θ going away from the origin are not harmless. When these figures are combined in the full three dimensional type space, we gain an extra degree of freedom as we can add a constant to the value for each allocation without changing the incentives, resulting in the harmless set illustrated in part (d).

Multi-Agent Mechanisms

As further motivation for our characterizations of the implementability of all deterministic single-agent mechanisms in Theorem 2, in this section, we discuss how to identify an agent's harmless set of types in multi-agent scenarios by leveraging on the results presented in the previous sections. Essentially, this boils down to a three step process:

1. Choose a truthful multi-agent mechanism $\mathcal{M}_{ma}$.
2. Derive a set of corresponding single-agent allocation rules F_{sa} which are implementable with payments.
3. Characterize $H(\theta, F_{sa})$ for each single agent.

To illustrate this process, consider a scenario with n unit-demand agents and two items, i_1 and i_2. Given this, the set of possible assignments for each agent is $S = \{\emptyset, i_1, i_2\}$. Assume that the mechanism $\mathcal{M}_{ma}$ is the incentive compatible Vickrey-Clarke-Groves (VCG) auction (Vickrey 1961; Clarke 1971; Groves 1973) that allocates items to agents such that the social welfare is maximised and charges each agent her externality. Thus, the allocation and the payment of each agent depends on the types reported by the other agents. The next step is to derive the corresponding single-agent allocation rules F_{sa}. In the case of VCG, every single-agent implementable-with-payments allocation rule f_{sa} is characterized by prices p_1 and p_2 for item i_1 and i_2, respectively, which correspond to the agent's externality. f_{sa} then assigns the agent the item (or nothing) she prefers at those prices. One of the f_{sa} characterized by prices p_1 and p_s is shown in Figure 3(a) where it is possible to observe that if the agent's type is in the red area no item is allocated to the agent, if it is in the blue area then she gets item i_1, and if it is in the green area she gets item i_2. Without restrictions on the types of the other agents, every non-negative pair of prices p_1 and p_2 is possible, and thus this defines the set of single-agent allocation rules F_{sa}. Because every pair of prices is possible, we can immediately apply Theorem 2 for the final step to characterize the harmless set.[4]

[4]Strictly speaking VCG is a family of mechanisms determined by tie-breaking rules, our results apply to identify the set of types that is simultaneously harmless for all tie-breaking rules.

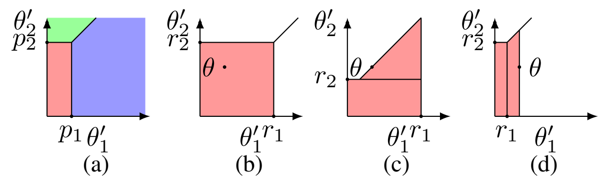

Figure 3: (a) Example of allocation rule f_{sa} of single-agent mechanisms induced by a multi-agent mechanism where no item is allocated to the agent whose type is in the red area, if it is in the blue area then she gets item i_1, and if it is in the green area she gets item i_2. (b, c, d) Examples of how the Harmless set of types changes for different values of reserve prices r_1 and r_2 set by the mechanism.

Corollary 2. *Let F denote the set of implementable-with-payments deterministic allocation rules and let F_{sa} denote the set of single-agent allocation rules derived from VCG. Then $H(\theta, F) = H(\theta, F_{sa})$.*

This three step process can be applied to *any* truthful multi-agent mechanism. In general, step 2 is an application of the taxation principle, and step 3 follows the logic of the proof of Theorem 2. Some cases, such as affine maximizers with finite agent weights and zero allocation weights (Roberts 1979), yield the same result as VCG, but others are more complex. For example, in the same setting as before but with additional reserve prices r_1 and r_2, not every set of prices is possible, because r_1 and r_2 serve as lower bounds. Thus, the harmless set of types depends also on the specific value of r_1 and r_2 as shown in Figure 3 (b,c,d).

Allocation-Dependent Verification

We have largely assumed that the set of verifiable types depends only on the true type. Some authors assume, however, that the set of verifiable types also depends on the allocation received. For example, in the combinatorial auction setting studied by Fotakis, Krysta and Ventre 2014, they assume that the mechanism designer can only determine ex post whether the agent over-reported her value for the assignment she received. (See the extended version of the paper for a discussion and example of how our approach can be applied in this case).

Reverse Approach

Our tools can be used also to answer the following question: given a reported type $\theta' \in \Theta$ and a class of mechanisms, what types need to be verified? In this case, the verification aims to check if a type $\hat{\theta} \in \Theta$ is the true type of the agent. Thus, from the perspective of the mechanism designer it is unnecessary to verify whether $\hat{\theta}$ is the agent's true type, if an agent with true type $\hat{\theta}$ cannot benefit by reporting θ'. We call the types that need to be verified *harmful*.

Definition 11. *Given a reported type θ' and an allocation rule f, the* harmful set of types *$Z(\theta', f)$ is the set composed* *by the types $\hat{\theta} \in \Theta$ such that $f(\theta') \cdot \hat{\theta} > f(\hat{\theta}) \cdot \hat{\theta}$.*

In the extended version of the paper, we show how to straightforwardly adapt our formulation to harmful sets of types. As mentioned before, we have chosen to present our primary approach as identifying the harmless set of types given a set of agents, allocation rules, and agents' true types because it leads to appealing geometrical characterization. The reverse approach is more natural for direct application by a mechanism designer because it is directly phrased in terms of what to do for a given report. In particular, the steps the designer has to follow to use verification as a substitute for money are the following. First, the designer decides which family of implementable-with-payments allocation rules to use and collects the agents' reported types. Then he verifies that each agent's true type is not in the set of types $Z(\theta', F)$ and, if necessary, penalizes the agents by, e.g., excluding them in the allocation. Finally, he applies the chosen allocation rule. The downside of the reverse approach is that the geometric characterization is more complex. In the end however, the two are equivalent as all that matters is identifying the set of (θ, θ') pairs for which verification is needed.

Application Examples

We conclude with two additional applications. For k-minded combinatorial auctions, we show that can recover previous results about when a particular verification is or is not sufficient and that we can extend them by characterizing a verification that would be sufficient for the case where it is not. For K-facility location, we show how our framework allows us to recover a sufficient verification for a particular class of mechanisms and extend it to a much larger class.

k-Minded Combinatorial Auctions

Consider the (known) k-minded combinatorial auction setting studied by Fotakis, Krysta and Ventre 2014. In this setting a set of goods must be allocated to a set of agents, and an agent has some value for exactly k subsets of them. (More precisely, she receives some set of items and her utility is that of the most valuable of the k sets of which they are a superset.) They showed that for $k = 1$, all implementable-with-payments allocation rules are also implementable using a verification that prevents agents from overbidding, while for $k > 2$ this is not the case. This result follows easily from our results, that also provide a nice visual intuition for what goes wrong in the $k = 2$ case.

For $k = 1$, from a single agent perspective there are effectively two possible assignments, $S = \{s_1, s_2\}$: the agent does not get her desired bundle $\theta_{s_1} = 0$ or she does and gets value θ_{s_2}. From Theorem 1 (deterministic mechanisms) or Theorem 4 (randomized mechanisms), we see that the harmless types are exactly those where the agents underbids. Thus, being able to verify the agent did not overbid suffices.

For $k = 2$, we simply add a new assignment s_3. Letting $\theta_2 = \theta_{s_3}$ and $\theta_1 = \theta_{s_2}$, Figure 1(a) shows the harmless set for deterministic mechanisms with the s_1 dimension omitted as $\theta_{s_1} = 0$. Again applying Theorem 1, the harmless set no longer includes all types where the agent underbids. In the example shown, the agent prefers s_3 to s_2, and so types where the agent underbids on s_3 but underbids more on s_2 are not harmless. Thus, this is exactly the sort of misreport that makes being able to verify that the agent has not overbid insufficient. It also shows that a sufficient verification is that the agent has correctly reported her value for her preferred assignment and not overreported her value for the other assignment. Whether this verification is reasonable or not depends on the application.

For randomized mechanisms, the technical Lemma used to prove Theorem 4 (see the extended version of the paper) can be directly applied to yield the following theorem.

Theorem 5. *Let θ be such that $\theta_{s_i} \neq \theta_{s_j}$ for all i and j and $m \geq 3$. The harmless set of all single agent truthful in expectation mechanisms with $\theta_{s_1} = 0$ is $\{\lambda\theta : \lambda \leq 1\}$.*

Note that, in contrast to Theorem 4, adding a constant to the value of each non-null allocation is no longer harmless because it changes values relative to the null allocation.

K-Facility Location

Consider a set of G potential locations where a set of K facilities will ultimately be located ($|G| > |K|$). Agents will be assigned to one of the K facilities and have preferences over the facility they are assigned to. In particular their utility for being assigned to the facility at location $g \in G$ is $\theta_g = b - c_g$ where b is the benefit of using a facility and c_g is the cost associated with using the facility at location g. We study the resulting mechanism design problem under the assumption that the mechanism can enforce the assignment of an agent to a particular facility, an assumption called *cluster imposing* in the literature (Ferraioli, Serafino, and Ventre 2016). We can directly apply Theorem 2 to characterize the verification needed to ensure that all deterministic implementable-with-payments allocation rules are truthful with this verification. At this level of abstraction, Figure 1(a) captures the relevant pairwise constraints, and the overall harmless set is not substantively different than in our analysis of VCG (which uses an implementable-with-payments allocation rule for this problem) except that the null assignment is not permitted.

Our results become more interesting when we study the restricted case where the agents and possible locations are on a line and c_g is the distance from the agent's location to g. This setting was previously studied by Ferraioli et al. (2016), who showed that in addition to the cluster imposing assumption, a combination of two (allocation dependent) verifications is sufficient to implement every *efficient* deterministic mechanism (with fixed tie-breaking). The first, *no underbidding*, ensures the agent cannot report that she is closer to her assigned facility than she actually is. The second, *direction imposing*, ensures the agent cannot report she is to the left of her assigned facility when she is actually to the right (and vice versa). Because agent locations are restricted to be on the line, agent types are quite restricted. When restricting to the pairwise case, if (WLOG) the agent prefers the right location, the harmless set for all implementable-with-payments allocation rules consists of all types to the left of the agent along the line. If the agent's location is in between the two possible facility locations, then their two verifications exactly cover the complement of the harmless set: no underbidding prevents reports to the right of the agent's location but left of the facility location while direction imposing prevents reports to the right of the facility location. If the agent is located to the right of both facilities, neither verification prevents misreports further to the right. Instead, the restriction to allocation rules which use fixed tie-breaking ensures that these reports never change the allocation, so the harmless set in this case is actually the entire space.

In addition to providing an intuitive illustration of why their verifications are sufficient (and in a sense necessary as well), we can strengthen their characterization to cover a larger class of mechanisms. In particular, let a *fixed tie-breaking implementable-with-payments allocation rule* be an implementable-with-payments allocation rule with the additional property that all types which are indifferent between two allocations at prices implied by the allocation rule receive the same allocation.

Corollary 3. *In the cluster imposing case, the no underbidding and direction imposing verifications suffice to implement all (efficient and approximate) fixed tie-breaking implementable-with-payments allocation rules.*

We can also shed more light on whether their verifications are necessary. They show that eliminating any one of them breaks truthfulness, which our results succintly illustrated. However, their verifications are stronger than necessary in that they are still applied in the case where the agent would already receive her preferred allocation by reporting truthfully (and so the harmless set is the entire space). So in principle the verifications could be weakened to no-underbidding-when-not-receiving-preferred-allocation and direction-imposing-when-not-receiving-preferred-allocation respectively.

References

Ashlagi, I.; Braverman, M.; Hassidim, A.; and Monderer, D. 2010. Monotonicity and implementability. *Econometrica* 78(5):1749–1772.

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

Dughmi, S., and Ghosh, A. 2010. Truthful assignment without money. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, EC '10, 325–334.

Ferraioli, D.; Serafino, P.; and Ventre, C. 2016. What to verify for optimal truthful mechanisms without money. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, 68–76.

Fotakis, D., and Zampetakis, E. 2015. Truthfulness flooded domains and the power of verification for mechanism design. *ACM Trans. Econ. Comput.* 3(4):20:1–20:29.

Fotakis, D.; Krysta, P.; and Ventre, C. 2014. Combinatorial auctions without money. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, 1029–1036.

Fotakis, D.; Krysta, P.; and Ventre, C. 2015. The power of verification for greedy mechanism design. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, 307–315.

Frongillo, R., and Kash, I. 2014. General truthfulness characterizations via convex analysis. In *International Conference on Web and Internet Economics*, 354–370. Springer.

Gibbard, A. 1973. Manipulation of voting schemes: A general result. *Econometrica* 41(4):587–601.

Green, J. R., and Laffont, J.-J. 1986. Partially verifiable information and mechanism design. *Review of Economic Studies* 53(3):447–456.

Groves, T. 1973. Incentives in teams. *Econometrica: Journal of the Econometric Society* 617–631.

Guesnerie, R., and Laffont, J.-J. 1984. A complete solution to a class of principal-agent problems with an application to the control of a self-managed firm. *Journal of public Economics* 25(3):329–369.

Guo, M., and Conitzer, V. 2010. Strategy-proof allocation of multiple items between two agents without payments or priors. In *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3*, 881–888.

Koutsoupias, E. 2014. Scheduling without payments. *Theor. Comp. Sys.* 54(3):375–387.

Krysta, P., and Ventre, C. 2015. Combinatorial auctions with verification are tractable. *Theor. Comput. Sci.* 571:21–35.

Penna, P., and Ventre, C. 2009. Optimal collusion-resistant mechanisms with verification. In *Proceedings 10th ACM Conference on Electronic Commerce (EC-2009), Stanford, California, USA, July 6–10, 2009*, 147–156.

Procaccia, A. D., and Tennenholtz, M. 2009. Approximate mechanism design without money. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, EC '09, 177–186.

Roberts, K. 1979. The characterization of implementable choice rules. *Aggregation and revelation of preferences* 12(2):321–348.

Saks, M., and Yu, L. 2005. Weak monotonicity suffices for truthfulness on convex domains. In *Proceedings of the 6th ACM conference on Electronic commerce*, 286–293. ACM.

Satterthwaite, M. A. 1975. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory* 10(2):187 – 217.

Serafino, P., and Ventre, C. 2014. Heterogeneous facility location without money on the line. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, 807–812.

Serafino, P., and Ventre, C. 2016. Heterogeneous facility location without money. *Theor. Comput. Sci.* 636(C):27–46.

Ventre, C. 2014. Truthful optimization using mechanisms with verification. *Theor. Comput. Sci.* 518:64–79.

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