The Revelation Principle for Mechanism Design with Signaling Costs

ANDREW KEPHART and VINCENT CONITZER, Duke University, USA

The *revelation principle* is a key tool in mechanism design. It allows the designer to restrict attention to truthful mechanisms, greatly facilitating analysis. This is also borne out algorithmically, allowing certain computational problems in mechanism design to be solved in polynomial time. Unfortunately, when not every type can misreport every other type (the *partial verification* model), or—more generally—misreporting can be costly, the revelation principle can fail to hold. This also leads to NP-hardness results. The primary contribution of this paper consists of characterizations of conditions under which the revelation principle still holds when reporting can be costly. (These are generalizations of conditions given earlier for the partial verification case [11, 21].) Furthermore, our results extend to cases where, instead of reporting types directly, agents send signals that do not directly correspond to types. In this case, we obtain conditions for when the mechanism designer can restrict attention to a given (but arbitrary) mapping from types to signals without loss of generality.

CCS Concepts: • **Theory of computation → Algorithmic mechanism design**;

Additional Key Words and Phrases: automated mechanism design, signaling costs, partial verification, revelation principle, evidence, costly misrepresentation

ACM Reference Format:

Andrew Kephart and Vincent Conitzer. 2020. The Revelation Principle for Mechanism Design with Signaling Costs. 1, 1 (December 2020), 39 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Mechanism design concerns making decisions based on the private information of one or more agents, who will not report this information truthfully if they do not see this to be in their interest. The goal is to define a *mechanism*—a game to be played by the agent(s)—that defines actions for the agents to take and maps these actions to outcomes, such that in equilibrium, the agents' true types map to outcomes as desired.

This may appear to leave us in the unmanageable situation of having to search through all possible games we could define. Fortunately, we are rescued by the *revelation principle*. It states that anything that can be implemented by a mechanism can also be implemented by a *truthful* mechanism, where agents in equilibrium report their types truthfully.

Authors' address: Andrew Kephart, andrewkephart@gmail.com; Vincent Conitzer, conitzer@cs.duke.edu, Duke University, Department of Computer Science LSRC Building D101 308 Research Drive, Durham, North Carolina, USA, 27708-0129.

Manuscript submitted to ACM

This vastly reduces the search space, and under certain circumstances allows optimal mechanisms to be found in polynomial time (e.g., if the number of agents is constant and randomized outcomes are allowed [6, 7]). And of course, the revelation principle is also extremely useful in the process of obtaining analytical results in mechanism design.

Unfortunately[1], there are situations where the revelation principle fails to hold. Notably, this is the case in settings with *partial verification*.[2] Here, some types may be unable to report that they are other types (without detection). For example, an agent in an online marketplace may be able to pretend to arrive between 0min and 30min after she really does, but not any earlier or later.

We can use this example to show how non-truthful mechanisms can sometimes allow us to distinguish types better than truthful mechanisms. An agent of type (i.e. arrival time) 12:10 can report that it is of type '12:20', but not of type '12:45'. Thus, an agent of type 12:20 can prove that it is not of type 12:10 by reporting that it is of type '12:45'. In contrast, a truthful report of type '12:20' does not prove this, because a type 12:10 agent can generate such a report as well.

Due to this absence of a revelation principle in this context, the problem of determining whether a particular choice function can be implemented becomes, in general, NP-hard [1]. It is thus natural to seek to characterize those conditions on the misreporting graph—which represents which types can misreport which other types— under which the revelation principle holds. Such a characterization can help us obtain analytical results in mechanism design and algorithms for efficiently finding mechanisms. Indeed, these conditions have been previously characterized [11, 21]; we review them later as they come up as special cases of our characterizations.

In practice, it is not always black and white whether one type can misreport another. Often, one type can misreport another *at some cost*. We call this *mechanism design with reporting costs*. Reporting costs may correspond to financial expense or to expended effort. For example, a minor may acquire a fake driver's license at some price. A consumer may engage in tricks to improve his credit score. In college admissions, a student may improve on his/her "natural" SAT score by extra prepping for the test.[3] Generally, for every type $\theta \in \Theta$ and every report $\hat{\theta} \in \Theta$ that the agent might report, there is a non-negative cost $c(\theta, \hat{\theta})$ for doing so (with $c(\theta, \theta) = 0$). Traditional mechanism design is the special case where $c(\theta, \hat{\theta}) = 0$ everywhere; partial verification is the special case where $c(\theta, \hat{\theta}) \in \{0, \infty\}$ and $c(\theta, \theta) = 0$. Because partial verification is a special case, it immediately follows that the revelation principle does not hold *in general* with costly reporting. However, in this paper, we identify necessary and sufficient conditions for the revelation principle to still hold, generalizing the earlier conditions for the partial revelation case [11, 21].

Furthermore, we present our results for a more general setting,[4] namely the one where agents may be restricted to send (costly) signals that do not necessarily correspond directly to types. We call this *mechanism design with signaling costs*. Here we say an agent *emits* a signal s, rather than *reports* a type θ. This puts us in the domain of signaling games. Consider, for example Spence's famous model of education [19]. In this model, agents signal their type (capabilities) by attaining a certain level of education; the idea is that higher-ability agents can more easily attain a higher level of education, so that in equilibrium the different types separate and employers take the level of education into account in hiring decisions. In this case, there is no *ex-ante* correspondence between types and signals.

[1] This is only unfortunate from analytical and algorithmic viewpoints. Indeed, often a non-truthful mechanism in these settings will perform better than any truthful one. We show several examples of this.
[2] Also known as *hard evidence*.
[3] We have in mind here not prepping that has value beyond the test—e.g., studying algebra in order to be able to solve more problems—but rather acquiring tricks—e.g., about how best to guess when unsure—that improve the score on the test but are otherwise of no societal value.
[4] The EC'16 version of this paper did not consider this more general setting.

Now consider an employer who can commit to a mapping from signals (levels of education) to hiring decisions. This employer then is in a position to design a mechanism, but cannot restrict attention to "truthful" mechanisms in a straightforward sense, because it is not clear what reporting truthfully would mean. It would be very helpful to the employer to know a mapping from types to signals (that the agent would be incentivized to follow) with the following property: if there is *any* such mapping that suffices for the employer's objective, then so does this one. In standard mechanism design, that special mapping is the truthful one. In our case, it is not clear which mapping, if any, works. But, for any given mapping, we provide necessary and sufficient conditions for it to have the desired property. This, of course, generalizes the case discussed before where signals correspond to types and reporting truthfully is costless; there the mapping of interest is the truthful one.

We believe that the mechanism design with signaling costs framework will be of increasing importance. This is because those running mechanisms increasingly have data on the agents, as opposed to knowing nothing about them *ex ante* and only being able to ask them about their preferences.

Now, an agent can often change the data that the mechanism has about it. But this can come at some effort (or other) cost to the agent. Hence, the standard mechanism design framework where an agent can report any type at no cost—the "anonymous bidder walking into a Sotheby's auction" model[5]—does not exactly fit such applications. But the mechanism design with costly signaling framework does.

In Sections 6 and 7, we consider cases where aspects beyond the signaling costs, such as the valuation function and choice function, are known.

1.1 Introductory Example - Inspection Game

The "Inspection Game" example illustrates our model as well as the failure, in general, of the revelation principle in the mechanism design with reporting costs setting.[6] [7] We will introduce another, more complex example in Section 3.

Suppose Beth is considering buying a crate of produce from Pam. The produce can be fresh, decent, or rotten (and Pam will know which it is). Beth can either accept or reject the crate. If the produce is fresh or decent, Beth would like to accept it, otherwise she would like to reject it. This gives:

$\Theta = \{fresh, decent, rotten\}$ — the set of types.
$O = \{accept, reject\}$ — the set of outcomes.
$F = \{fresh \to accept, decent \to accept, rotten \to reject\}$ — the choice function Beth seeks to implement.

Before making her decision, Beth can inspect the produce. However, at some cost Pam can add dyes or scents to the produce, which will alter how it appears to Beth. Thus we also have:
$S = \{\hat{fresh}, \hat{decent}, \hat{rotten}\}$ – the set of signals.

[5]Of course, the standard mechanism design framework where misreporting is costless can perfectly well address situations where the party running the mechanism has prior information over the agent. The point is that the standard framework does *not* address the agent being able to *change* this prior information at some cost.
[6]We present a similar example in [15]. This is the smallest example where a choice function is implemented by some non-truthful mechanism, but not by any truthful one.
[7]Of course, we are not the first to note that the revelation principle may not hold when reports are limited. [11] show this for the special case of partial verification.

Since we are in the reporting costs setting, this set is the same as the set of types. We add the " ˆ " symbol to each signal to distinguish it from its corresponding type. The following matrix gives the cost of making a crate of type θ give off the signal s. For example, at a cost of 30, Pam can make a crate of rotten produce appear fresh. Since we are in the reporting costs setting, $c(\theta, \hat{\theta}) = 0$

$c(\theta, s) =$

	$\hat{fresh}$	$\hat{decent}$	$\hat{rotten}$
fresh	0	0	0
decent	10	0	0
rotten	30	10	0

The value that Pam receives if the crate is accepted is 20:[8]

$v_\theta(accept) = 20$
$v_\theta(reject) = 0$

Beth needs to commit to a mechanism for choosing an outcome based on how the produce appears. The naïve mechanism of accepting the produce whenever it does not appear rotten, $H = \{\hat{fresh} \rightarrow accept, \hat{decent} \rightarrow accept, \hat{rotten} \rightarrow reject\}$ would fail. It would be worth it to Pam to pay the cost of 10 to make rotten produce appear to be decent, netting 10 value. Hence Beth would end up inadvertently accepting rotten produce.

What Beth should do instead is use $N = \{\hat{fresh} \rightarrow accept, \hat{decent} \rightarrow reject, \hat{rotten} \rightarrow reject\}$. Under N, if the produce really is rotten it will not be worth it for Pam to alter it, and it will be rejected. On the other hand, if the produce is decent, Pam will make it appear to be fresh and it will be accepted, resulting in the implementation of Beth's desired choice function.

Here the revelation principle does not hold. There exists a set of outcomes, choice function, and valuation function (namely those given here) such that there exists a mechanism (namely N) that implements the choice function. But, there exists no mechanism (H would be our best candidate) that both incentivizes the agent to report truthfully and implements the choice function.

This example had the property that signals are type reports, and reporting one's type is costless. In the more general setting where types emit signals that do not directly correspond to the types, it is not immediately clear what truthful reporting means; in this context, we will be interested in whether we can, without loss of generality, restrict our attention to some mapping G from types to signals. In the example above, the mapping of interest was $G = \{fresh \rightarrow \hat{fresh}, decent \rightarrow \hat{decent}, rotten \rightarrow \hat{rotten}\}$, and we showed that we cannot restrict attention to it without loss of generality.

1.2 Intuition Behind Main Results

We can capture a high-level intuition behind our main results (in the reporting costs setting):

The revelation principle holds
if and only if

[8] In general, the value of each outcome to the agent can depend on the type.

$$a\hat{c} \text{ is small relative to } a\hat{b} \text{ and } b\hat{c}$$

for all types a, b, c.
(Recall that $\hat{c}$ is the signal corresponding to the type c. Here we also introduce shorthand $a\hat{c}$ for the cost of a emitting $\hat{c}$.)

Why is this the case? Let us try to create a choice function F violating the revelation principle. Thus, F should be non-truthfully implementable, but not truthfully implementable.

Construct F such that a envies b's outcome o_b over its own, o_a. Thus, in a truthful mechanism, when $a\hat{b}$ *is small*, a will want to emit $\hat{b}$ to receive o_b. Hence F is not truthfully implementable. When $b\hat{c}$ *is also small*, we can try to non-truthfully implement F by having b emit $\hat{c}$ to receive o_b.

But, if $a\hat{c}$ *is small*, a would emit $\hat{c}$ as well and thus we would not be able to implement F. Hence the revelation principle holds. On the other hand, if $a\hat{c}$ *is large*, we can successfully keep o_b away from a at $\hat{c}$, implementing F. So the revelation principle fails.

1.2.1 Relation to Inspection Game. We illustrate how our intuition applies to the inspection game example. Consider the type triple: $a = rotten$, $b = decent$, and $c = fresh$. The heuristic for this triple is:

$$\text{The revelation principle holds}$$
$$\text{if and only if}$$
$$c(rotten, \hat{fresh}) \text{ is small relative to } c(rotten, \hat{decent}) \text{ and } c(decent, \hat{fresh}).$$

In the inspection game $c(rotten, \hat{fresh}) = 30$, $c(rotten, \hat{decent}) = 10$, and $c(decent, \hat{fresh}) = 10$. These costs violate the heuristic's inequality. [9] Hence the revelation principle does not hold.

But, if we were to modify the cost function to respect the heuristic's inequality, we can observe that the revelation principle holds.

- Let $c(rotten, \hat{fresh}) = 0$. Under N it's now worth it for Pam to make rotten fish appear fresh. So Beth would accept rotten fish under N. Hence N no longer implements F, so the revelation principle holds.
- Let $c(rotten, \hat{decent}) = 30$. Under H it's not worth it for Pam to make rotten fish appear decent. So Beth would reject rotten fish under H. Hence H now implements F, so the revelation principle holds.
- Let $c(decent, \hat{fresh}) = 30$. Under N it's not worth it for Pam to make decent fish appear fresh. So Beth would reject decent fish under N. Hence N no longer implements F, so the revelation principle holds.

2 MODEL

As is common in this type of setting, we focus on the case of a single *signal-emitting* agent; this corresponds to holding the other agents' signals fixed.

[9] Here we are deliberately vague as to the precise mathematical statement of the inequality as it can vary based on the setting we are considering. We will give precise statements later on.

In an *instance*, we have a set of *types* Θ; we will generally use $\theta, \theta_1, \theta_2, \ldots$ to denote variable types and $a, b, c, \ldots$ to denote specific types. We have a set of *signals* S; with $s, s_1, s_2 \ldots$ denoting variable signals and x, y, z denoting specific signals. (When agents report types directly, we have $S = \Theta$.)

We then have the *revelation principle mapping* $G : \Theta \to S$ which designates a signal for each type. The question is whether the designer can restrict attention, without loss of generality, to mechanisms in which each type θ emits signal $G(\theta)$. (When agents report types directly, the mapping G of interest is always the identity function, corresponding to truthful reporting.)[10]

We assume throughout that G is a one-to-one (injective) mapping from types to signals and only maps types to signals which have a finite cost for that type. This automatically holds in the type-reporting case. In the general signaling model, it is not entirely without loss of generality,[11] but if G maps two types to the same signal then they can never receive distinct outcomes.

This assumption allows us to overload our notation by using $\theta, \theta_1, \ldots, a, b, \ldots$ also to indicate the designated signal of the type with the same name. That is, we use a shorthand where θ refers both to the type θ and the signal $G(\theta)$. This is natural in the type-reporting setting and remains convenient in the general signaling setting. Because of this, we will rarely mention G explicitly (it is generally held fixed), but G is implicit in how the signals are named.

We also have a set of *outcomes* (alternatives) O. There is a *valuation function* $v : \Theta \times O \to \mathbb{R}$, where $v_\theta(o)$ is the valuation that type θ has for outcome o.

Finally, there is a *cost function* $c : \Theta \times S \to \mathbb{R}_{\geq 0}$, where $c(\theta, s)$ denotes the cost type θ incurs when emitting s. We often use the shorthand ax for $c(a, x)$. Combining this shorthand with that for G, we will often use ab to mean $c(a, G(b))$.

A *mechanism* is defined by, first, an *allocation function* $A : S \to O$, where $A(s) = o$ denotes that the mechanism chooses outcome o when signal s is emitted. When we allow for transfers, then another part of the mechanism is the *transfer function* $T : S \to \mathbb{R}$, where $T(s)$ denotes the transfer *received* by the agent when emitting s. (Hence, $T(s) < 0$ implies the agent is making a payment.) The agent's utility for having type θ, emitting signal s, and receiving outcome o and transfer t is $u(\theta, s, o, t) = v_\theta(o) - c(\theta, s) + t$.

Let $R : \Theta \to S$ denote a *response* for the agent to the mechanism, where $R(\theta) = s$ denotes that the agent emits s when her true type is θ. We say R is *optimal* for mechanism $M = (A, T)$ if for all θ and s, $u(\theta, R(\theta), A(R(\theta)), T(R(\theta))) \geq u(\theta, s, A(s), T(s))$.

We are generally interested in *implementing* a *choice function* $F : \Theta \to O$. We sometimes use the shorthand o_a for $F(a)$. A mechanism $M = (A, T)$ together with response R *implements* F if R is optimal for M and for all θ, $A(R(\theta)) = F(\theta)$. (Moreover, it implements it with transfer $T(R(\theta))$ and utility $u(\theta, R(\theta), A(R(\theta)), T(R(\theta)))$ for type θ.)

We say a mechanism is *truthful* if all types emit in accordance with G, and *non-truthful* otherwise. Similarly, we say a type θ *misemits* if it emits a signal other than $G(\theta)$.

[10] The next section will discuss the revelation principle mapping in more detail.
[11] For example, consider a setting with two types and only one possible signal. Here, there is only one possible mapping G and so, technically, the revelation principle holds for this mapping, even though it maps two types to the same signal.

We will sometimes use N to denote a not-necessarily truthful mechanism and H to denote a truthful one.[12] Additionally, R_N and R_H will refer to optimal responses for the respective mechanisms.

2.1 Truthfulness and The Revelation Principle Mapping, G

Essential to the Revelation Principle is a clear understanding of *truthfulness*. In the type-reporting setting, truthfulness is straightforward: a type is truthful if it reports itself. In the signaling setting, things are not so simple. Because there is no assumed relationship between signals and types, we have no inherent notion of truthfulness.[13] Rather than imposing an arbitrary definition of truthfulness, we make it user-defined with the G mapping. (With the "user" being the mechanism designer.) Thus, "truthfulness" can be whatever is natural for the instance at hand.

We do enforce one restriction on G: that it be one-to-one from types to signals. This is for two reasons:

- A one-to-one G mapping is (implicitly) assumed in earlier revelation principle work in the type-reporting setting: under truthful reporting, obviously, distinct types are supposed to make distinct type reports. By continuing this assumption, our work is more directly comparable.
- We believe that revelation principles for G functions that are not one-to-one would look significantly different and make the characterizations significantly more cumbersome without much benefit. This is because when $G(\theta_1) = G(\theta_2)$, any choice function F with $F(\theta_1) \neq F(\theta_2)$ will automatically not be truthfully implementable according to the definition.[14] Thus, the revelation principle holds if and only if none of these choice functions are implementable non-truthfully (and, additionally, the other conditions hold). Proving this would be much different in flavor than the proofs for the one-to-one case that we present here. Our proofs study whether non-truthful implementations can be 'transformed' into truthful implementations, rather than whether they exist at all.

2.2 Revelation Principle

We say that the revelation principle (RP) holds on a mapping G whenever we can, without loss of generality, restrict our attention to truthful mechanisms when trying to implement the choice function.

Definition 2.1 (Revelation Principle). Given Θ, S, and c, the *revelation principle holds on G* if:

> For every O, v, $N = (A_N, T_N)$, and R_N which is optimal for N.
>
> There exists another mechanism $H = (A_H, T_H)$ with $A_H(G(\theta)) = A_N(R_N(\theta))$ for all θ, where the truthful-emitting response R_H (with $R_H(\theta) = G(\theta)$) is optimal for H.

Hence, any choice function F that can be implemented can be implemented truthfully.

[12] Here, H stands for "honest" to prevent confusion with the transfer function T.
[13] It is tempting to define a type's truthful emission as the signal which has the lowest emission cost for it, but this is limiting. We can imagine situations in which everyone can send a common signal of 'doing nothing' at zero cost, but emitting the natural 'truthful' signal comes at some cost, such as filling out a form.
[14] The same issue occurs if G is stochastic—the agent is expected to use a mixed strategy—and multiple types place positive probability on the same signal. This is one reason why we only consider pure strategies for the agent.

Here both the valuations and choice function are unknown when we determine whether the revelation principle holds. This is the standard way of interpreting the revelation principle.

Later we will consider a variation on the revelation principle where valuations are known. Finally we will consider a further variation where both the valuations and choice function are known, which we call a *fully specified instance*.

Sometimes, we will also wish to implement the choice function with specific transfers and/or utilities.

Definition 2.2 (Fixed Transfers). The revelation principle holds with *fixed transfers* (to the agent) if the revelation principle holds when we add the additional constraint on the mechanism H that:

$T_H(G(\theta)) = T_N(R_N(\theta))$ for all θ.

Definition 2.3 (Fixed Utilities). The revelation principle holds with *fixed utilities* if the revelation principle holds when we add the additional constraint on the mechanism H that:

$u(\theta,\ G(\theta),\ A_H(G(\theta)),\ T_H(G(\theta))) = u(\theta,\ R_N(\theta),\ A_N(R_N(\theta)),\ T_N(R_N(\theta)))$ for all θ.

If transfers or utilities are not fixed, we say they are *variable*.

We also consider scenarios without transfers, of which there are two flavors:

Definition 2.4 (No Transfers In Equilibrium). The revelation principle holds with *no transfers in equilibrium* if it holds when we only consider the space of (truthful and non-truthful) mechanisms M where:

$T_M(R_M(\theta)) = 0$ for all θ.

Note that this allows negative (or positive, but those would not be useful) transfers to signals the agent does not emit.

Definition 2.5 (No Transfers At All). The revelation principle holds with *no transfers at all* if it holds when we only consider the space of mechanisms M where:

$T_M(\cdot) = 0$.

Note that if the revelation principle holds with fixed transfers, then it also holds when we wish to have no transfers in equilibrium. This does not automatically imply that the revelation principle holds when we have no transfers at all. The difference between the two no-transfers conditions becomes relevant when we have a signal that we would prefer the agent never use. With no transfers in equilibrium, we can put a sufficiently negative transfer on it to prevent use. But, with no transfers at all, our only recourse would be to put an unappealing outcome on it, which might not be sufficient.

But, as we will show, in the type-reporting setting, with unknown valuations the revelation principle is the same, and with known valuations it is essentially the same. Finding useful conditions that ensure the revelation principle holds with no transfers at all when we are in the signaling setting is currently an open problem.

We will use acronyms to refer to our various revelation principles. The possible words used to describe a revelation principle and the letter(s) used to represent the words are as follows: **F**ixed; **V**ariable; **T**ransfers; **U**tilities; **N**o **T**ransfers (in) **E**quilibrium; **N**o **T**ransfers **A**t **A**ll; **K**nown (valuations). For instance, *FTFU* means fixed transfers and fixed utilities; *KNTEVU* means known valuations, no transfers in equilibrium, and variable utilities.

2.2.1 Finding the G Mapping. In this work we assume the G mapping is given exogenously. This is certainly helpful if there is a natural candidate for G. The obvious case is when the type space and the signal space are equal, and G corresponds to reporting truthfully.

Sometimes, the type and signal spaces will differ, but there still is a natural G. For example, suppose that the agent's private information is her native language. To send a signal, she can take any one of a number of written language tests. The natural G mapping would be to expect her to take the test corresponding to her native language.

But this might not always be the best way to proceed. For example, the native language may have a script that is hard to master, so that passing the agent's own language test comes at significant cost. Expecting the agent to instead take the test for a related language with a simpler script may help in implementation. We may even use languages not in the type space. E.g., a Latin test could pick out speakers of a Romance language, even though today there are no native Latin speakers.

Yet, in spite of these theoretical possibilities, we might suspect the revelation principle holds: if something can be implemented at all, then it can also be implemented by a mechanism for which the agent has an incentive to take the test in her native language. Our results allow one to determine when this is the case.

However, in general, our conditions do not give any explicit guidance as to when a G satisfying the revelation principle exists, and if so, how to find it. We leave this to future research.

3 RUNNING EXAMPLE: STOCKING FOOD BANKS

We will use a running example of stocking food banks to illustrate the various instantiations of our framework. This example is type-reporting, as the signal space and the type space coincide. Imagine that a city has four districts: *North*, *East*, *South*, *West*, each of which has a food bank and a population of people living in it. A person's type is the district where she lives.

Based on demographics and health conditions, the city government has determined how much it values the population of each district receiving certain types of food. For example, it may wish to distribute milk in a district with many young children, and vegetables in a district with many single, middle-aged people. Note that the city's objective is different from maximizing the sum of the utilities of the people who will make use of the food bank. In this example, we assume there are three food types, (those high in) *fiber*, *protein*, and *vitamins*.

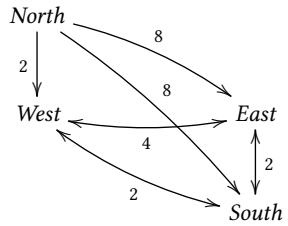
Determining which food to stock in each bank would be straightforward except that the population from one district might travel to another if they prefer the latter's food. This would correspond to them "misreporting" their district.

Such misreporting is costly because it requires traveling to another district. These costs are summarized as follows:

$c(\theta, \hat{\theta}) =$

	North	*West*	*East*	*South*
North	0	2	8	8
West	∞	0	4	2
East	∞	4	0	2
South	∞	2	2	0

It can also be useful to visualize this reporting cost structure as a graph. The directed edge between two districts represents the cost of traveling from one to the other. We leave off the zero-cost self reporting edges.[15] All other edges not shown are assumed to have infinite cost.



Suppose that the people of each district value the food as follows.

$v_\theta(o) =$

	fiber	*protein*	*vitamins*
North	1	9	1
West	2	3	3
East	1	3	6
South	1	1	1

Moreover, suppose that the city's objective function is as follows.

$J(\theta, o) =$

	fiber	*protein*	*vitamins*
North	10	0	0
West	0	10	5
East	0	10	5
South	0	5	10

The city's problem is to specify a mechanism—that is, which food each district's food bank distributes—such that it is happy with the equilibrium result. Possibly, the city also can distribute a (welfare) transfer to each person who comes to the food bank.

We assume that there are no capacity constraints on any food bank—that is, it does not run out of food if too many people come. Hence, this is a mechanism design problem for a single agent—the person turning up at the food bank—because other agents' decisions do not affect this agent.

There are multiple variants of this problem, depending on whether the city can make transfers, whether it wants these transfers to be a certain amount, whether it cares about the transportation costs incurred by people traveling to a different district, etc.

If the revelation principle holds for the variant in question, the city's problem is significantly easier; it can focus on truthful implementations, i.e., mechanisms such that nobody would travel to another district. If it does not hold, the city needs to consider mechanisms that do incentivize some districts' populations to travel to the food bank in another

[15] In the signaling setting these edges might not be zero-cost.

district, because this may result in better outcomes than any truthful mechanism can achieve. We will see examples of both cases.

4 SUMMARY OF REVELATION PRINCIPLE RESULTS

We now summarize our main results, those for the revelation principle with unknown valuations and unknown choice function, i.e. the standard revelation principle.

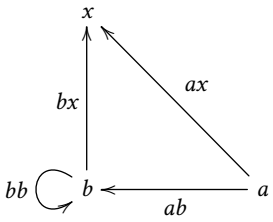
4.1 Intuition High Level Overview

We first provide a high level overview of the intuition behind our main results.

The revelation principle holds iff for any choice function F implemented by a non-truthful mechanism N, there also exists a truthful mechanism H which implements F. That is, for every θ we have $H(\theta) = F(\theta)$ and $R_H(\theta) = \theta$. For the purpose of providing intuition, we limit ourselves here to the case where only one type signals non-truthfully in N. Under N let b obtain $F(b)$ by emitting some signal $x \neq b$.

Under H, $A(b) = F(b)$ so b can now obtain $F(b)$ by emitting truthfully. This risks that some type a now prefers emitting b to truthfully emitting a.

We can visualize these types and the relevant signaling costs between them as follows[16]. We use our shorthand for the cost function here, so that (for instance) ax is the cost incurred by an agent of type a to emit the signal x.



This way of displaying the costs makes heavy use of the fact that we conflate types and the signals those types emit to, which is natural in the type-reporting case but a bit more subtle in the general signaling case. An edge in this graph corresponds to the type at the beginning of the edge emitting the signal at the end of the edge, and the weight on the edge is the cost for doing so. In general, some signals in the graph may not have a type associated with them; such signals cannot have outgoing edges in the graph. In the type-reporting setting, every vertex has a self-edge with cost 0, but this is not true in the general signaling setting.

There are two ways to use transfers to keep a truthful in H.

Variable Transfers, Variable Utilities
First: Pay an agent extra for emitting a. This keeps a honest but may cause other types to misemit to a under H. So we

[16]Other types not shown in the graph may exist.

pay off all the types which can emit a too, and so on. This works as long as the process terminates without reaching b. If we had to pay b, the extra transfers would cancel out and all would be for naught. Thus, with variable transfers and variable utilities, if there exists no path of finite emission cost edges along signals in $range(G)$ from b to a, the revelation principle holds. (Note that signals not in $range(G)$ do not have a type corresponding to them and thus by definition cannot have outgoing edges.)

Variable Transfers, Fixed and Variable Utilities
Second: Pay less to an agent for emitting b. We can safely subtract $bx - bb$ from the transfer that b received for emitting x under N (making b equally happy under H and N).[17] Note that a did not emit x under N. Consider how much greater a's incentive is to emit b under H than to emit x under N. It is $(ax - ab) - (bx - bb)$, which will be nonpositive iff $ax \leq ab + bx - bb$. In this case a still will not misemit. Thus, with variable transfers, if $ax \leq ab + bx - bb$ holds, then the revelation principle holds with fixed utilities (and thus with variable utilities as well). Note that in the case where $bb = 0$ this condition is equivalent to the triangle inequality holding on the signaling cost structure.

Fixed Transfers, Variable Utilities
If we have fixed transfers (but variable utilities), neither of the previous two techniques are allowed. So, a should prefer not emitting b outright, given it did not want to emit x under N. Ensuring this requires $ab \geq ax$. Additionally, this condition is only necessary when b can actually emit x.[18] So we have $bx < \infty \implies ab \geq ax$.

Here, our limited example leaves out an aspect that will affect the general condition. The additional aspect is that a might emit some signal $y \neq a$ in the non-truthful mechanism.[19] Hence an honest emission of a by a would leave it $aa - ay$ worse off in the truthful mechanism, which affects a's incentives to misemit to b. To keep this from happening, (when $bx < \infty$) we now need $ab - (aa - ay) \geq ax$, i.e. $ab - aa \geq ax - ay$.

Fixed Transfers, Fixed Utilities
Finally, with fixed transfers and fixed utilities, b must receive the same utility in both H and N. This requires that there exists some δ_b such that when $bx \neq \infty$ then $bx = bb = \delta_b$. This then reduces to the partial verification setting and we can use the known revelation principle for that case. In our notation this becomes: $(ab = \delta_a \wedge bx = \delta_b) \implies ax = \delta_a$.

As it turns out, our reasoning above captures all the key aspects, so our general results involve the exact same conditions as those presented here (though the proofs are more intricate).

4.2 Summary of Main Results

The revelation principle holds with {variable, fixed} transfers and {variable, fixed} utilities *iff* for all ordered doubles $a \neq b$ of types and all ordered doubles x, y of signals:

[17] Subtracting any more might cause b to misemit under H.
[18] We don't need to explicitly check this for any of the other cases as bx is already present in the conditions.
[19] This aspect does not play a part in the variable transfers revelation principles as we can compensate a for any change in signaling costs.

	Variable Utilities	Fixed Utilities
Variable Transfers	$ab + bx - bb \geq ax$, *or* no b to a path[21]	$ab + bx - bb \geq ax$
Fixed Transfers	$bx < \infty \implies ab - aa \geq ax - ay$	$bx \in \{\delta_b, \infty\}$, *and* $(ab = \delta_a \wedge bx = \delta_b) \implies ax = \delta_a$

where $0 \leq \delta_a, \delta_b < \infty$.

To recover the revelation principles for the type-reporting case set bb, aa, ay[22], δ_a, and δ_b to 0.

And we can restate for the signaling setting the intuition from Section 1.2 as:

The revelation principle holds
if and only if
ax is small relative to ab and bx

for all types a, b and every signal x.

4.2.1 Triangle Inequalities. In both variable transfers cases, if $bb = 0$ then the inequality conditions become triangle inequalities. This corresponds to real-world scenarios where smaller misrepresentations are proportionally more costly than larger misrepresentations.

The case where the type corresponds to the agent's geographic location (and emission cost to travel effort) fits this criterion especially well, which is why we chose it for our running example. The effort of traveling from location x to location z will almost always[23] be no more than adding the costs of traveling from x to y and then y to z. Thus, when a government or company wants to determine outcomes based on geographic location (and has flexibility with giving out transfers), it is almost always sufficient to use a truthful mechanism.

4.3 Related Work

In earlier work, we studied the computational complexity of deciding whether a given choice function can be implemented when misreporting is costly [15]; these results will be relevant in Section 7 of this paper.

Several other papers study the revelation principle in the setting of partial verification, including the more general variant where the signal space differs from the type space (but we still have $c(\theta, s) \in \{0, \infty\}$ for all θ, s). [4] and [9] consider what [16] calls instances satisfying *normality*. Here, each type has some "maximal evidence" signal it can emit, which does as much to distinguish it from other types as any other signal can. The revelation principle holds for the specific mapping where each type emits its maximal evidence.

With signaling costs, though, normality is not as helpful. Particularly, the maximal evidence for a type might be so costly that it would never consider emitting it.

[21] Along signaling cost edges between signals in $range(G)$.
[22] $ay = 0$ in the type-reporting case as this is always the minimum possible value for it, which is when $y = a$.
[23] This is not *always* the case, e.g., due to discontinuities in mass-transit pricing. For instance, a ten-hour train ride may require an overnight sleeper car, which would cost more than two five-hour train rides.

When normality does not hold, both [4] and [9] consider dynamic mechanisms of the following form. First, the agent gives a costless type report. Then, the mechanism requests she follow up with a 'verifying' signal or signals which are the same as those she would emit in the non-truthful implementation. If the agent does not emit the verifying signal, she is allocated a punishment outcome. Building on these ideas, [20] proposes a reinterpretation of the revelation principle. If we consider the verifying signal to be part of outcome, then the only 'reporting' happening is a costless type report. Hence, in a sense, the revelation principle still holds.

We find this dynamic approach unsatisfactory in our context. It is of little use to the designer in the search through the space of mechanisms, because it does not help in determining which signal each type should eventually (after the costless type report) emit.[24] And that search corresponds to an NP-hard problem [1]. In contrast, if we know which signal each type is supposed to emit, the problem becomes easy. This is what the revelation principle does for us in the traditional mechanism design setting, and what we would like any revelation principle to reproduce here.[25]

A variety of papers consider revelation type principles in more limited partial verification or costly signaling settings. These include: [17] for contracts in an exchange economy; [3] for enforcement of contract disputes; [14] for when agents have preferences for honesty; [16] for when each type has a signal which can be used to distinguish it from all other types. [10] shows that when costs are monotonically increasing with a signal's distance from 'truth', increasing the number of signals available for agents to use expands the set of implementable choice functions. Additionally, they characterize the optimal mechanism for a setting with a one-dimensional type space.

Other papers explore implementability of choice functions in costly signaling-like settings, but do not derive revelation principles. [13] gives a necessary condition, *evidence-monotonicity*, which is required for a choice function to be implementable with no signaling costs incurred in equilibrium. [5] characterizes truthfully implementable choice functions in the setting of *probabilistic verification*, in which there is a certain *probability* that a lying agent will be caught.

We also consider our work related to machine learning in contexts where what is being classified is an agent that may put in some effort to resist accurate classification. The most obvious version of this is *adversarial classification*: detecting spam, intrusions, fraud, etc. when the perpetrators wish to evade detection [2, 8]. However, as the examples in the introduction indicate, there are many situations where the objectives of the classifier and agent are not diametrically opposed. This is also brought out in more recent work on "strategic classification," [12] which has a heavier focus on the machine learning aspect.

5 RESULTS: REVELATION PRINCIPLE

5.1 Variable Transfers, Fixed Utilities

Consider the case where, given a non-truthful mechanism, we wish to obtain a truthful mechanism that implements the same choice function and maintains the utility that each type obtains, but can differ in the transfers made to the agent.

[24] It is somewhat more helpful when a non-truthful mechanism might involve multiple rounds of signaling as is the case in [4] and [9].
[25] One caveat is that, in the general signaling setting, we do not provide an algorithm for finding the without-loss-of-generality mapping from types to signals.

Definition 5.1 (VTFU Condition). An instance satisfies the *VTFU condition* if for all ordered doubles $a \neq b$ of types, for every signal x,

$$ax \leq ab + bx - bb$$

Note that if $bb > ab + bx$, the VTFU condition cannot hold as ax will always be non-negative.

Theorem 5.2. *The RP holds with variable transfers and fixed utilities iff the VTFU condition holds.*

Proof.

VTFU condition $\implies$ *RP holds*

We will show that for any F, for any non-truthful implementation, we can construct a truthful implementation. Let N together with R_N implement F. Let $H = N$ except:

$A_H(\theta) = F(\theta)$ for $\theta \in \Theta$

$T_H(\theta) = T_N(R_N(\theta)) - \theta R_N(\theta) + \theta\theta$ for $\theta \in \Theta$

If H is truthful, it clearly implements F with fixed utilities. We show this is the case.

The following series of inequalities holds for any $a \neq b$, and $x = R_N(b)$:

$$\begin{aligned}
& v_a(o_a) + T_H(a) - aa \\
&= v_a(o_a) + T_N(R_N(a)) - aR_N(a) && \text{by definition of } T_H \\
&\geq v_a(o_b) + T_N(x) - ax && \text{by optimality of } R_N \\
&\geq v_a(o_b) + T_N(x) - bx - ab + bb && \text{by VTFU condition} \\
&= v_a(o_b) + (T_H(b) + bx - bb) - bx - ab + bb && \text{by definition of } T_H \\
&= v_a(o_b) + T_H(b) - ab
\end{aligned}$$

Similarly, for any type a and signal $z \notin range(G)$ we have:

$$\begin{aligned}
& v_a(o_a) + T_H(a) - aa \\
&= v_a(o_a) + T_N(R_N(a)) - aR_N(a) && \text{by definition of } T_H \\
&\geq v_a(A_N(z)) + T_N(z) - az && \text{by optimality of } R_N \\
&= v_a(A_H(z)) + T_H(z) - az && \text{by definition of } H
\end{aligned}$$

This shows that under H, a is willing to emit a over any other signal. Thus, H is truthful.

VTFU condition not holding $\implies$ *RP does not hold*

Choose an instance where the VTFU condition is violated, i.e., there exist $a \neq b \in \Theta$ and $x \in S$ such that:

$$ax > ab + bx - bb$$

We first consider the cases where x is equal to a or b:

– If $x = b$, then $ab > ab + bb - bb$, which clearly cannot be the case.

– If $x = a$, then $aa > ab + ba - bb$, i.e. $aa - ab > ba - bb$. Consider the choice function $F(\cdot) = o$ for some arbitrary outcome o. F is clearly implementable by some (possibly non-truthful) mechanism.

  But F is not truthfully implementable by any mechanism H. To keep a from misemitting to b, we need $T_H(a) - T_H(b) \geq aa - ab$. And, to keep b from misemitting to a, we need $ba - bb \geq T_H(a) - T_H(b)$. But this leads to a contradiction as $aa - ab > ba - bb$. Hence the revelation principle fails.

So, from now on we assume $x \neq a, b$.

Define λ s.t. $ax > \lambda > ab + bx - bb$.

Choose an arbitrary outcome o and let $F(\cdot) = o$. N defined by:

$A_N(\cdot) = o$
$T_N(x) = \lambda$
$T_N(a) = aa$
$T_N(\theta) = -bb$ for $\theta \neq x, a$

implements F and is non-truthful as b will prefer to emit x over itself as $v_b(o) + \lambda - bx > v_b(o) + ab - bb \geq v_b(o) - bb \geq v_b(o) - 2bb = v_b(o) + T(b) - bb$. And since $ax > \lambda$, a will emit truthfully over emitting x and thus obtain a utility of $v_a(o)$.

Consider any mechanism H where for all θ, $A_H(\theta) = o$, and truthful emission gives it the same utility it achieved under N. Thus:

$T_H(b) \geq \lambda - bx + bb$, and
$T_H(a) = aa$

If the revelation principle is to hold, this mechanism must be truthful. But in fact, by emitting b, a can obtain a utility of:

$$\begin{aligned} & v_a(o) + T_H(b) - ab \\ \geq\ & v_a(o) + \lambda - bx + bb - ab && \textit{by definition of } T_H(b) \\ >\ & v_a(o) && \textit{by definition of } \lambda \end{aligned}$$

which is what it would get for emitting a. So the revelation principle does not hold. □

5.1.1 Revisiting the Running Example. Suppose the city would like to maximize its objective and have equal utilities for all types. [26] The best truthful mechanism is the following, resulting in an objective of 30 while ensuring each type achieves a utility of exactly 3. [27]

[26] Because the city can make arbitrary transfers, any mechanism where all types receive utility u_0 can easily be transformed into another mechanism where all types receive utility u_1 and that implements the same choice function, simply by adding $u_1 - u_0$ to each transfer.
[27] Again, this utility can easily be transformed into any other number.

$H =$

	$\hat{North}$	$\hat{West}$	$\hat{East}$	$\hat{South}$
A	*fiber*	*vitamins*	*protein*	*protein*
T	2	0	0	2

But might there be a non-truthful mechanism that achieves a higher objective? When we check the VTFU condition, we see that the following triples violate it:

a = North, b = West, x = East
a = North, b = West, x = South

Thus, the revelation principle does not hold. And in fact there is a better non-truthful mechanism. Consider the following non-truthful mechanism under which *West* travels to the *South* food bank. It results in an objective of 35 while still giving all types utility 3.

$N =$

	$\hat{North}$	$\hat{West}$	$\hat{East}$	$\hat{South}$
A	*fiber*	*fiber*	*protein*	*protein*
T	2	0	0	2

5.2 Variable Transfers, Variable Utilities

We consider the case where the transfers and utilities the agent achieves can differ between non-truthful and truthful implementations. That is, all that is needed for the revelation principle to hold is that for any non-truthful implementation of a choice function, there exists a truthful implementation as well.

Definition 5.3 (VTVU Condition). An instance satisfies the *VTVU condition* if for all ordered doubles $a \neq b$ of types and every signal x, either:

(i) There exists no path from b to a of finite signaling cost edges between signals in $range(G)$, or

(ii) $ax \leq ab + bx - bb$.[28]

Theorem 5.4. *The RP holds with variable transfers and variable utilities iff the VTVU condition holds.*

Proof.
VTVU condition $\implies$ *RP holds*
Consider the signaling graph consisting of the vertices in $range(G)$ and the edges with finite signaling costs between signals in $range(G)$. Consider the strongly connected components of this graph; the graph decomposes into a DAG over these components.

Suppose there is a mechanism N that, together with a response R_N, non-truthfully implements a choice function F. Let us restrict our attention to types inside a single component. For these types, for *any* signal (even one outside the component), part (ii) of the VTVU condition holds since there clearly is a path between every pair of types in the component. Thus the VTFU condition holds on the types within this component. Hence, if we considered a restricted

[28]Note that (ii) is identical to the VTFU condition.

instance consisting of only the types in this component (and all signals) the RP holds with fixed utilities by Theorem 5.2. So, when we return to the unrestricted instance, this means that every type θ in the component can receive $F(\theta)$ and some transfer at θ, while also having no incentive to misemit to any other signal inside the component.

Choose such an internally truthful implementation (including transfers) with fixed utilities within each component. For signals not in any component (i.e., signals not in $range(G)$), let the implementation be the same as N. There will be no incentive to misemit inside each component by internal truthfulness. Nor will there be incentive to misemit to a signal that is outside every component because the utility each agent receives is the same as in N . But, this does not necessarily result in a mechanism that is truthful overall, because there may be incentives to misemit across components. We can fix this as follows.

We order the components in a way consistent with the DAG, such that types in a later component can emit types in an earlier component, but not vice versa. Additionally, for each component we specify an additional transfer that all types in that component will receive. By making this transfer sufficiently larger for later components in this order, no type will wish to misemit to an earlier component (and misemitting to a later component comes at infinite cost).[29]

Specifically, if component C_1 comes before C_2, then the additional transfers $\pi^{\text{add}}_{C_1}$ and $\pi^{\text{add}}_{C_2}$ should be such that for all $\theta_1 \in C_1$ and $\theta_2 \in C_2$,

$$v_{\theta_2}(F(\theta_2)) + \pi^{\text{orig}}_{\theta_2} + \pi^{\text{add}}_{C_2} - \theta_2\theta_2 \geq v_{\theta_2}(F(\theta_1)) + \pi^{\text{orig}}_{\theta_1} + \pi^{\text{add}}_{C_1} - \theta_2\theta_1$$

$$\Leftrightarrow$$

$$\pi^{\text{add}}_{C_2} - \pi^{\text{add}}_{C_1} \geq v_{\theta_2}(F(\theta_1)) + \pi^{\text{orig}}_{\theta_1} - \theta_2\theta_1 - v_{\theta_2}(F(\theta_2)) - \pi^{\text{orig}}_{\theta_2} + \theta_2\theta_2$$

where the π^{orig} are the transfers obtained from applying the VTFU revelation principle within the components.

Since the additional transfer to a component is the same for all types in it, internal truthfulness is maintained. Since the additional transfers are positive no type will want to emit a signal that is outside every component. So, since no type will misemit within a component, to a signal outside every component, or to another component, the implementation is truthful.

VTVU not holding $\implies$ *RP does not hold*
Given that the VTVU condition does not hold, there exists some $a \neq b$ and x with:

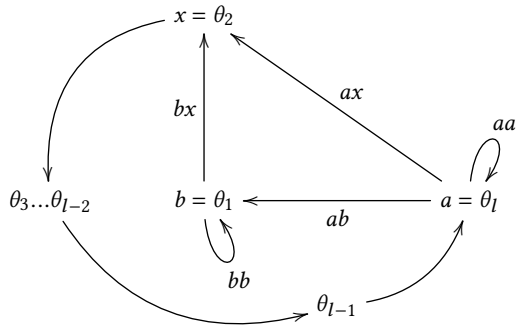
$$ax > ab + bx - bb$$

for which there is a path of types connected by finite-cost edges: $b = \theta_1, \theta_2, \ldots, \theta_{l-1}, \theta_l = a$ (where we consider $l + 1 = 1$).

By the same reasoning as in this part of the proof of Theorem 5.2 (fixed rather than variable utilities) we know that when x equals a or b the revelation principle fails to hold. So, from now on we assume $x \neq a, b$.

If the path goes through x, we can assume without loss of generality that $x = \theta_2$. We can visualize this as follows.

[29] Note that if there are an infinite number of components, transfers may go to infinity in the later components.



Define λ s.t. $ax > \lambda > ab + bx - bb$.

Consider an outcome set with a separate outcome o_{θ_i} for every type θ_i (except $o_b = o_x$), and a valuation function with:

$v_{\theta_i}(o_{\theta_i}) = \theta_i\theta_i$
$v_{\theta_i}(o_{\theta_{i+1}}) = \theta_i\theta_{i+1}$
except,
$v_a(o_b) = \lambda$
$v_b(o_x = o_b) = bx$
and $v = 0$ elsewhere.

Consider the mechanism N where $A_N(\theta_i) = o_{\theta_i}$, except $A_N(b) = o_a$. Moreover, let T_N be a large constant value on all the θ_i and x, and 0 everywhere else, so that none of the θ_i would misemit somewhere outside that set.

Then an optimal response has $R_N(\theta_i) = \theta_i$, except $R_N(b) = x$, since:

- For any $\theta_i \notin \{a, b\}$, the only viable alternative is to misemit θ_{i+1}, but the cost of doing so exactly cancels out the benefit.

- For b, emitting x results in utility $T_N + bx - bx = T_N$, which is no worse than emitting truthfully and getting $T_N + v_b(o_a) - bb \leq T_N$.

- For a, emitting x would give utility $T_N + \lambda - ax$, which is less than the $T_N + aa - aa$ it gets for emitting truthfully.

Hence, this is a non-truthful implementation of some choice function with $F(\theta_i) = o_{\theta_i}$.

On the other hand, such a choice function cannot be implemented truthfully. This is because we have a Rochet-style negative cycle [18]. Truthful emissions require $T_H(\theta_i) \geq T_H(\theta_{i+1})$ for each θ_i, except:

$$
\begin{aligned}
T_H(b) + v_b(o_b) - bb &\geq T_H(x) + v_b(o_x) - bx, \text{ i.e.} \\
T_H(b) + bx - bb &\geq T_H(x) && \textit{by definition of } v_b \\
&\text{and} \\
T_H(a) + v_a(o_a) - aa &\geq T_H(b) + v_a(o_b) - ab, \text{ i.e.} \\
T_H(a) &\geq T_H(b) + v_a(o_b) - ab, \text{ i.e.} && \textit{by definition of } v_{\theta_i} \\
T_H(a) &\geq T_H(b) + \lambda - ab && \textit{by definition of } v_a \\
&> T_H(b) + (ab + bx - bb) - ab && \textit{by definition of } \lambda \\
&= T_H(b) + bx - bb \\
&\geq T_H(x) && \textit{by final equation from above}
\end{aligned}
$$

Hence $T_H(a) > T_H(x)$. But this leads to a contradiction as we follow the inequalities around the cycle.

For the case where the path does not go through x, we can make a similar argument, treating x as before but using a separate outcome o_{θ_2} for θ_2 and letting $v_b(o_{\theta_2}) = b\theta_2$. By doing so, b is indifferent between x and θ_2 in the non-truthful implementation which therefore still works[30], and for the cycle we have:

$$
\begin{aligned}
T_H(b) + v_b(o_b) - bb &\geq T_H(\theta_2) + v_b(o_{\theta_2}) - b\theta_2, \text{ i.e.} \\
T_H(b) + bx - bb &\geq T_H(\theta_2) && \textit{by definitions of } v_b \textit{ and } v_{\theta_i}
\end{aligned}
$$

allowing us to have $T_H(a) > T_H(\theta_2)$ and thus we get the same contradiction around the cycle. □

5.2.1 Relation to Partial Verification. Earlier work studied the revelation principle in the partial verification case.

Definition 5.5 (Partial Verification). An instance is a *partial verification instance* if:

- The signal set and the type set are the same.
- $\forall \theta : \ G(\theta) = \theta$
- $\forall \theta : \ \theta\theta = 0$
- $\forall \theta, s : \ \theta s \in \{0, \infty\}$

In the partial verification case, if we allow transfers and utilities to vary, it is known that the revelation principle is characterized by the following "Strong Decomposability" condition [21] in the reporting graph (where there is an edge from a to b if and only if a can report b, at zero cost):

(1) Every strongly connected component is fully connected, i.e., every type in the component can report every other type.

[30] So b still emits x.

(2) All types within the same strongly connected component have the same image set, i.e., the set of types they can report is the same.

We can observe what the VTVU condition simplifies to in the partial verification case.

Definition 5.6 (Partial Verification VTVU Condition). An instance satisfies the *partial verification VTVU condition (PVVTVU)* if for all ordered doubles of types $a \neq b$, and every type x, either:

(i) There exists no path from b to a of zero reporting cost edges, or

(ii) $ab, bx = 0 \implies ax = 0$

Proposition 5.7. *In the partial verification case, strong decomposability is equivalent to the PVVTVU condition.*

Of course, if both results are correct, this must in fact be the case. Nevertheless, it is instructive to verify it directly.

Proof.

The PVVTVU condition $\implies$ Strong Decomposability

First we show (1). Assume $a \neq b$ and x are in the same strongly connected component, and that $ab = 0$ and $bx = 0$. Since they are in the same component, a is reachable from b along a path with edges of zero reporting cost. Thus (i) of the PVVTVU condition does not hold, so (ii) must. By (ii), $ax = 0$. This implies transitivity, and hence full connectivity, within every strongly connected component.

Second we show (2). Suppose a and b are in the same strongly connected component — so by (1), $ab = 0$ — and $bx = 0$ for some x. Because a is reachable from b, (ii) of PVVTVU must hold. Thus $ax = 0$. Hence, nodes in a strongly connected component have the same image set.

Strong Decomposability $\implies$ The PVVTVU condition

Suppose $ab = bx = 0$ and a is reachable from b with edges of zero reporting cost; it suffices to show that $ax = 0$. We know a and b must be in the same strongly connected component, so by (2), $bx = 0 \implies ax = 0$.[31] □

5.2.2 Revisiting the Running Example. Suppose the city wants to maximize its objective without regard to the transfers or the resulting utilities. When we consider the conditions for the revelation principle we just obtained, we see that part (ii) of the VTVU condition is violated by the following triples: (*North, West, East*) and (*North, West, South*). However, in both cases there is no path back to *North*.

Hence, the revelation principle holds. Thus, any choice function that is implementable is implementable truthfully. So the city only needs to search through the space of truthful mechanisms to maximize its objective. The following truthful mechanism achieves the best possible objective value of 40.

$H =$

	$\hat{North}$	$\hat{West}$	$\hat{East}$	$\hat{South}$
A	*fiber*	*protein*	*protein*	*vitamins*
T	6	0	1	0

[31] The careful reader may wonder why we did not need to use condition (1) in this part of the proof. This is because condition (1) in Strong Decomposability is in fact redundant—condition (2) implies condition (1).

5.3 Fixed Transfers, Variable Utilities

We consider the case where, given a non-truthful mechanism, we wish to obtain a truthful mechanism that implements the same choice function and makes the same transfers to the agent, but can differ in the utility obtained by the agent.

Definition 5.8 (FTVU Condition). An instance satisfies the *FTVU condition* if for all ordered doubles $a \neq b$ of types, for all ordered doubles x, y of signals,

$$bx < \infty \implies ab - aa \geq ax - ay$$

Theorem 5.9. *The RP holds with fixed transfers and variable utilities iff the FTVU condition holds.*

Proof.

FTVU condition $\implies$ *RP holds*

We will show that for any F, for any non-truthful implementation we can construct a truthful implementation. Let N together with R_N implement F. Define $H = N$ except:

$A_H(\theta) = F(\theta)$ for $\theta \in \Theta$

$T_H(\theta) = T_N(R_N(\theta))$ for $\theta \in \Theta$

$T_H(z) = -L$ for some L large enough that no type would ever emit z, for $z \notin range(G)$.[32]

If H is truthful, it clearly implements F with fixed transfers. We show this is the case.

The following series of inequalities holds for any pair of types $a \neq b$, and signals $x = R_N(b)$[33], $y = R_N(a)$:

$$\begin{aligned}
& v_a(o_a) + T_H(a) - aa & \\
&= v_a(o_a) + T_N(y) - aa && \text{since } H \text{ and } N \text{ have fixed transfers} \\
&\geq v_a(o_a) + T_N(y) - ay + ax - ab && \text{by FTVU condition} \\
&\geq v_a(o_b) + T_N(x) - ax + ax - ab && \text{by optimality of } R_N \\
&= v_a(o_b) + T_N(x) - ab & \\
&= v_a(o_b) + T_H(b) - ab && \text{since } H \text{ and } N \text{ have fixed transfers}
\end{aligned}$$

For any type a, and signal $z \notin range(G)$ we have:

$$\begin{aligned}
& v_a(o_a) + T_H(a) - aa & \\
&\geq v_a(A_H(z)) + T_H(z) - az && \text{by definition of } T_H(z)
\end{aligned}$$

This shows that under H, a is willing to emit a over any other signal. Thus, H is truthful.

FTVU condition not holding $\implies$ *RP does not hold*

Let $a \neq b \in \Theta$ and $x, y \in S$ violate the FTVU condition. Thus:

$$bx < \infty, \text{ and } ab - aa < ax - ay$$

[32] This does not contradict our definition of fixed transfers as the definition allows transfers to signals that no type emits.

[33] This will imply $bx \neq \infty$.

Note that in the following proof, unlike those for variable transfers, we do not need to specially address when any of the following equivalencies (or pair of them) hold: $b = x,\ x = y$, or $y = a$. (Though if all three hold simultaneously, we have $a = b$, which is ruled out by definition).

We will show that we can define outcomes and valuation functions, as well as a non-truthful mechanism N (with an associated optimal response R_N) without transfers at all[34] that implement a choice function F that no truthful mechanism with fixed transfers would implement. That is, H with $A_H = A_N \circ R_N$ and $T_H(\theta) = 0$ for all θ is not truthful.

Create outcomes $o_a, o_b, \emptyset$ with:

$v_a(o_a) = ay$
$v_a(o_b) = ax$
$v_a(\emptyset) = 0$
$v_b(o_a) = 0$
$v_b(o_b) = bx$
$v_b(\emptyset) = 0$

Consider the mechanism N defined by $T_N = 0$, $A_N(y) = o_a,\ A_N(x) = o_b$, and otherwise $A_N(\cdot) = \emptyset$. Associated with it is some optimal response R_N for which $R_N(a) = y$ and $R_N(b) = x$.

However, a mechanism H with $T_H(a) = T_H(b) = 0$, $A_H(a) = o_a$, and $A_H(b) = o_b$ is not truthful. We have:

$$
\begin{aligned}
& v_a(o_b) - ab && \\
= {} & ax - ab && \text{by definition of } v_a \\
> {} & ay - aa && \text{by violation of the FTVU condition} \\
= {} & v_a(o_a) - aa && \text{by definition of } v_a
\end{aligned}
$$

Hence a would prefer to misemit b. □

5.3.1 Special Case: No Transfers In Equilibrium, Variable Utilities. We now consider the special case where we have no transfers in equilibrium. That is, the agent never receives any transfers (although we may allow transfers to signals no type would emits).

Of course, if the FTVU condition holds in general, then the revelation principle will also hold for this special case. However, we may wonder whether the full FTVU condition is still necessary, or if a more relaxed condition will do. It turns out that the full FTVU condition is still necessary.

Theorem 5.10. *The RP holds with no transfers in equilibrium and variable utilities iff the* FTVU condition *holds.*

Proof. For the *if* direction, no transfers in equilibrium is a special case of fixed transfers.

For the *only if* direction, in the proof of Theorem 5.9 we used a transfer function with no transfers at all, and thus no transfers in equilibrium. Hence the result carries over. □

[34]To prove the FTVU it is not necessary that N has no transfers at all. But, we will reuse this part of the proof in to prove Theorems 5.10 (no transfers in equilibrium) and 5.11 (no transfers at all), in which case having this condition is necessary.

5.3.2 Special Case: No Transfers At All, Variable Utilities, Type-Reporting Setting. We now consider the special case where we are in a type-reporting setting and have no transfers at all; that is, transfers are fixed at zero.

There is not necessarily any relation between this special case and the case of fixed transfers to the agent,[35] but as it turns out, the revelation principle will be the same.

Theorem 5.11. *The RP holds with no transfers at all and variable utilities in the type-reporting setting iff the* FTVU condition *holds.*

Proof. For the *if* direction, the proof of Theorem 5.9 derived a truthful mechanism with fixed transfers to types and large negative transfers on unused signals. In the type-reporting setting, under the truthful mechanism there are no unused signals. Hence the large negative transfers are no longer needed. So, for any non-truthful mechanism with no transfers at all we can create a corresponding truthful mechanism with no transfers at all.

For the *only if* direction, in the proof of Theorem 5.9 we used a transfer function with no transfers at all. Hence the result carries over. □

5.3.3 Special Case: No Transfers At All, Variable Utilities, Signaling Setting. Finding useful conditions that ensure the revelation principle holds when we are in the signaling setting, have no transfers at all, and variable utilities, is currently an open problem.

5.3.4 Revisiting the Running Example. In this setting, the city no longer cares that the types all receive the same utility, but now the city is unable to make welfare transfers. That is, transfers are fixed at zero. The *FTVU condition* does not hold for this cost function, the following assignments to a, y, b, and x all violate the condition:

a = *North*, y = *North*, b = *West*, x = *East*
a = *North*, y = *West*, b = *West*, x = *East*

a = *North*, y = *North*, b = *West*, x = *South*
a = *North*, y = *West*, b = *West*, x = *South*

a = *West*, y = *West*, b = *South*, x = *East*
a = *East*, y = *East*, b = *South*, x = *West*

Thus, we may wonder whether a non-truthful mechanism exists that is better than any other mechanism. However, it turns out that the revelation principle still holds *for the agent's specific valuation function* that we are considering here, and so in fact we can restrict attention to truthful mechanisms. We will return to this example in 6.6.1, at which point we will have given conditions for the revelation principle to hold for specific valuation functions in the FTVU case.

5.4 Fixed Transfers, Fixed Utilities

We consider the case where, given a non-truthful mechanism, we wish to obtain a truthful mechanism that implements the same choice function, makes the same transfers to each type, and maintains the utility that each type achieves.

[35] Indeed, in the setting of known valuations the conditions will turn out to differ.

Definition 5.12 (FTFU Condition). An instance satisfies the *FTFU condition* if there exists some $0 \leq \delta_\theta < \infty$ for each type θ such that for all pairs of type b and signal x,

$$bx \in \{\delta_b, \infty\}$$

and for all ordered doubles $a \neq b$ of types, and signal x,

$$(ab = \delta_a \wedge bx = \delta_b) \implies ax = \delta_a$$

Indeed, in the partial verification case (which, in our notation, is the type-reporting case where for all $a \neq b \in \Theta : ab \in \{0, \infty\}$), the condition $(ab = 0 \wedge bx = 0) \implies ax = 0$ is known as the *nested range condition* [11], which is known to characterize when the revelation principle holds in that case, with no transfers at all. Note that utilities are also necessarily fixed because no agent will ever incur nonzero reporting costs.

We first provide intuition as to why the FTFU condition is essentially the same as the nested range condition even though we have: a general signaling rather than type-reporting setting; fixed transfers rather than no transfers at all; and signaling costs that may be δ_θ rather than 0.

- In the nested range condition proof, that x may have a type corresponding to it does not play a role. Thus, we can extend it to a signaling setting with a, b as types and x as a signal.
- The key difference between implementations with fixed transfers and with no transfers at all is the ability to give a large negative transfer to a signal that no type emits. This is equivalent to banning its emission.

  This ban is useful for a 'nuisance' signal for which putting any outcome at it (without a negative transfer) would cause at least one type to misemit to it. Since we have fixed utilities, the presence of a nuisance signal is equally problematic in both truthful and non-truthful implementations. If a type emitted it under one implementation, it also would in the other. Thus the ability to ban it will not change when the revelation principle holds.
- Finally, for each type b, replacing 0 by δ_b shifts the utility function by a constant and does not affect behavior.

Theorem 5.13. *The RP holds with fixed transfers and fixed utilities iff the FTFU condition holds.*

Proof.
FTFU condition $\implies$ *RP holds*
We will show that for any F, for any non-truthful implementation we can construct a truthful implementation. Let N together with R_N implement F. Define $H = N$ except:
$A_H(\theta) = F(\theta)$ for $\theta \in \Theta$
$T_H(\theta) = T_N(R_N(\theta))$ for $\theta \in \Theta$

If H is truthful, it clearly implements F with fixed transfers and fixed utilities. (Note that because of the FTFU condition, the reporting cost for a type is always the same.) So now we show this is the case.

The following series of inequalities holds for any pair of types $a \neq b$, and signals $x = R_N(b)$, $y = R_N(a)$:

$$\begin{aligned}
& v_a(o_a) + T_H(a) - aa & \\
&= v_a(o_a) + T_N(y) - ay && \textit{since } H \textit{ and } N \textit{ have fixed transfers and utilities} \\
&\geq v_a(o_b) + T_N(x) - ax && \textit{by optimality of } R_N \\
&\geq v_a(o_b) + T_H(b) - ab && \textit{by definition of } T_H \textit{ and the FTFU condition}
\end{aligned}$$

For any type a, signal $y = R_N(a)$, and signal $z \notin range(G)$ we have:

$$\begin{aligned}
& v_a(o_a) + T_H(a) - aa & \\
&= v_a(o_a) + T_N(y) - ay && \textit{since } H \textit{ and } N \textit{ have fixed transfers and utilities} \\
&\geq v_a(A_N(z)) + T_N(z) - az && \textit{by optimality of } R_N \\
&= v_a(A_H(z)) + T_H(z) - az && \textit{by definition of } H
\end{aligned}$$

This shows that under H, a is willing to emit a over any other signal. Thus, H is truthful.

FTFU condition not holding $\implies$ *RP does not hold*
Consider first the case where for all $s \in S$, $s\theta \in \{\delta_\theta, \infty\}$, but there exist some $a \neq b$ and x such that $ab = \delta_a$, $bx = \delta_b$, and $ax = \infty$.

Let N be a mechanism with $A_N(\cdot) = o$, with o being any fixed outcome, and $T_N(\cdot) = 0$, except $T_N(x) = 1$. We have $R_N(b) = x$, so $T_N(R_N(b)) = 1$. Also, we know that $T_N(R_N(a)) = 0$ since a cannot emit x.

Now consider any mechanism H with $A_H = A_N \circ R_N$ and $T_H = T_N \circ R_N$. We then have $T_H(b) = 1$ and $T_H(a) = 0$. Since we also have $A_H(a) = A_H(b) = o$ and $aa = ab$, a would emit b over a and thus H is not truthful.

Hence there exists no truthful mechanism with the same transfers as N that implements the choice function. So the revelation principle fails.

Now, all that is left to do is to show that if for some b there is no δ_b (i.e., $bx \neq bb$, and $bx \neq \infty$ for some $b \in \Theta$ and $x \in S$) then the revelation principle fails.

Consider a mechanism N where A_N maps all signals to some fixed outcome o and T_N pays $bx + 1$ to any type that emits x, and 0 otherwise. For an optimal response R_N we have $R_N(b) = x$, so that b's utility is $1 + v_b(o)$.

Now consider any truthful mechanism H with $A_H(b) = o$ and where b receives the same utility as under N, which is $1 + v_b(o)$. $T_H(b)$ must be $bb + 1$ for this to be the case. But, since $bb \neq bx$ we will not have fixed transfers and thus the revelation principle fails. □

5.4.1 Revisiting the Running Example. Suppose the city is unable to make transfers and wants each type to receive the same utility. To make the example more interesting in this context, we add another outcome $\emptyset$, which consists of giving the agent nothing, for which all types have utility 0 and for which the city has objective 0. Clearly, the FTFU condition does not hold because there are edge costs that are neither 0 nor ∞. The optimal truthful mechanism in this

context is simply to give all agents $\emptyset$; this is simply because there is no value other than 0 that the types would all be able to get as their utility in this context. On the other hand, the following is an optimal non-truthful implementation (combined with the response where *West* and *East* misreport *South*, but *North* and *South* report truthfully), resulting in an objective value of 35 and each type obtaining utility 1.

$N =$

	$\hat{North}$	$\hat{West}$	$\hat{East}$	$\hat{South}$
A	*fiber*	$\emptyset$	$\emptyset$	*protein*
T	0	0	0	0

Now let us again modify the example, and replace the $\emptyset$ outcome with *cod liver oil*, which all types dislike (utility 0.1) but the city would love for the agent to take (objective 11 for all types). Obviously, in this case the optimal mechanism is to give all types *cod liver oil*, which is truthful. In this case, unlike in 5.3.4, what is happening is not that the revelation principle holds for the specific valuation function at hand, but rather that it holds for the specific combination of both the valuation function and the choice function at hand—that is, for the specific instance at hand. We will return to this in Section 7.

6 RESULTS: REVELATION PRINCIPLE FOR KNOWN VALUATIONS

So far, we have always considered whether the revelation principle holds for a given combination of Θ and $c : \Theta \times S \to \mathbb{R}_{\geq 0}$. For it to hold meant that *no matter what* the valuation function v and the choice function F (and, possibly, the transfer and/or utilities to be achieved) are, it is either truthfully implementable or not implementable at all.

But if we have a particular valuation function in mind, we may not care whether the revelation principle holds for other valuation functions; we just want to know whether *for this valuation function* we can restrict our attention to truthful mechanisms. Accordingly, in this section, we consider the case where the valuation function is known (or fixed). Hence, an *instance* will consist not only of Θ, S and c, but also O and v.

Later, in Section 7, we will consider the case where *everything* is fixed, including the choice function (and, possibly, the transfer and/or utilities to be achieved).

6.1 Known Valuations, Variable Transfers, Variable Utilities

We consider the case where the valuation function is known and we allow the transfers and utilities the agent achieves to vary between non-truthful and truthful implementations.

Finding useful conditions that ensure that the RP holds here is currently an open problem. We conjecture that verifying whether the revelation principle holds is NP-complete.

6.2 Known Valuations, Variable Transfers, Fixed Utilities

We consider the case where the valuation function is known and given a non-truthful mechanism, we wish to obtain a truthful mechanism that implements the same choice function and maintains the utility that each type obtains, but can differ in the transfers made.

With variable transfers and fixed utilities, it makes no difference whether we know the valuation function.

Theorem 6.1. *The RP holds with known valuations, variable transfers, and fixed utilities iff the VTFU condition holds.*

Proof.
VTFU condition holds $\implies$ *RP holds*
By Theorem 5.2, the VTFU condition implies that the RP holds for all possible valuation functions, so it will continue to hold for a specific valuation function.

VTFU condition not holding $\implies$ *RP does not hold*
In the corresponding part of the proof of Theorem 5.2 (without known valuations), we used a constant choice function whose choice of outcome did not matter. Hence, this part of the proof carries over unmodified to this case. □

6.3 Known Valuations, Fixed Transfers, Variable Utilities, Type-Reporting Setting

We consider the case where the valuation function is known, we are in a type-reporting setting and given a non-truthful mechanism, we wish to obtain a truthful mechanism that implements the same choice function and maintains the transfers paid to the agent, but can differ in the utility the agent obtains.

With fixed transfers and variable utilities in a type-reporting setting, it makes no difference whether we know the valuation function.

Theorem 6.2. *In the type-reporting setting the RP holds with known valuations, fixed transfers, and variable utilities iff the FTVU condition holds.*

Proof.
FTVU condition holds $\implies$ *RP holds*
By Theorem 5.9, the FTVU condition implies that the RP holds for all possible valuation functions, so it will continue to hold for a specific valuation function.

FTVU condition not holding $\implies$ *RP does not hold*
Let $a \neq b \in \Theta$ and $x, y \in \Theta$ violate the FTVU condition. Thus:

$$bx < \infty, \text{ and } ab - aa < ax - ay$$

Since we are in the type-reporting setting, we note that aa is zero. Additionally, since ay is always non-negative, we also have:

$$bx < \infty, \text{ and } ab < ax$$

We will show that for any set of outcomes and valuation function, we can define a non-truthful mechanism N (with an associated optimal response R_N) that implements a choice function F that no truthful mechanism with fixed transfers would implement. That is, H with $A_H = A_N \circ R_N$ and $T_H = T_N \circ R_N$ is not truthful.

We consider the following two cases separately:
(i) $bx \leq ba$ or $bx \leq ax$
(ii) $bx > ba$ and $bx > ax$

First consider (i):

Choose an arbitrary outcome o, and let $F(\cdot) = o$. Define N as follows:

$A_N(\cdot) = o$
$T_N(x) = bx + ax$
$T_N(a) = bx$
$T_N(\theta) = 0$ for $\theta \neq x, a$

Since $T_N(x) - T_N(a) = ax$, we can have a emit truthfully and receive bx. b will emit x over b since $T_N(x) > bx$, since $ax > ab \geq 0$. If $bx \leq ba$, b will emit x over a since $T_N(x) > T_N(a)$. If $bx \leq ax$, we can have b still emit x over a since:

$$\begin{aligned} & v_b(o) + T_N(x) - bx & \\ = \; & v_b(o) + ax & \textit{by definition of } T_N(x) \\ \geq \; & v_b(o) + bx & \textit{by our assumption that } bx \leq ax \\ \geq \; & v_b(o) + bx - ba & \textit{since signaling costs are always non-negative} \\ = \; & v_b(o) + T_N(a) - ba & \textit{by definition of } T_N(a) \end{aligned}$$

Hence N is non-truthful[36] and gives a transfer of $bx + ax$ to b and a transfer of bx to a.

For H to have fixed transfers we need $T_H(b) = bx + ax$ and $T_H(a) = bx$. If the revelation principle is to hold, this mechanism must be truthful. But, by emitting b, a can obtain a utility of:

$$\begin{aligned} & v_a(o) + bx + ax - ab & \\ > \; & v_a(o) + bx & \textit{by FTVU condition in type-reporting setting} \end{aligned}$$

which is what it would get for emitting a. So the revelation principle does not hold when $bx \leq ba$ or $bx \leq ax$.

Now consider (ii), which is that $bx > ba$ and $bx > ax$:

Choose an arbitrary outcome o, and let $F(\cdot) = o$. Define N as follows:

$A_N(\cdot) = o$
$T_N(x) = bx$
$T_N(\theta) = 0$ for $\theta \neq x$

N is non-truthful as a will emit x over a[37] since $T_N(x) = bx > ax$. And since $T_N(x) = bx$, we can have b emit truthfully and receive 0 transfer.

[36] The FTVU condition not holding implies $x \neq b$.
[37] (ii) implies $x \neq a$.

For H to have fixed transfers we need $T_H(b) = 0$ and $T_H(a) = bx$. If the revelation principle is to hold, this mechanism must be truthful. But in fact, by emitting a, b can obtain a utility of:

$$\begin{aligned} & v_b(o) + bx - ba \\ > \; & v_b(o) && \text{by our assumption that } bx > ba \end{aligned}$$

which is what it would get for emitting b. So the revelation principle does not hold when $bx > ba$ and $bx > ax$.

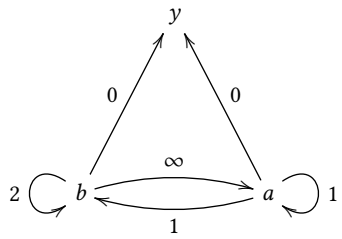
Since our two cases cover all the possibilities, the revelation principle does not hold. □

6.4 Known Valuations, Fixed Transfers, Variable Utilities, General Signaling Setting

We consider the case where the valuation function is known, we are in a general signaling setting and given a non-truthful mechanism, we wish to obtain a truthful mechanism that implements the same choice function and maintains the transfers paid to each type, but is allowed to differ in the utilities each type obtains.

Unlike with variable transfers and fixed utilities, the condition here is not the same as in the unknown valuations case (when we move beyond the type-reporting setting). Thus, we provide a separate condition for this case. It is not elegant, but can be checked in polynomial time.

6.4.1 Example: Difference in Revelation Principles for Known and Unknown Valuations. We first show that the FTVU condition is not necessary for the revelation principle to hold when we are in the signaling setting, have known valuations, fixed transfers, and variable utilities. (It is, of course, sufficient.) Consider an instance with two types $a \neq b$ and an additional signal y. Let $aa = 1,\ ab = 1,\ ay = 0,\ bb = 2,\ ba = \infty$, and $by = 0$. We can visualize these types and signaling costs as follows:



The FTVU condition does not hold on this instance. If we let $x = b$ we have:

$$ab - aa = 1 - 1 < 1 - 0 = ax - ay$$

Yet, we show with known valuations the revelation principle can still hold on this instance.

Suppose that we have just a single outcome that a and b both value at 0. When considering the design of truthful mechanisms, we need not worry about y, because we can put a sufficiently negative transfer there. Given this, it is easy to see that a mechanism H is truthful if and only if $T_H(a) \geq T_H(b)$. Hence, for the revelation principle to be violated, there must exist some non-truthful mechanism N where $T_N(R_N(b)) > T_N(R_N(a))$. But this leads to a contradiction no matter the choice of R_N:

- If both b and a emit y, then we cannot give them different transfers.
- If just a emits y (so b emits b), then for a to prefer emitting y to emitting b we need $T_N(y) \geq T_N(b) - 1$. But then b would also prefer to emit y.
- If just b emits y, if $T_N(y) > T_N(R_N(a))$, then a would also prefer to emit y.

Hence the revelation principle holds for these valuations.

We now move on to the revelation principle condition for this case.

6.4.2 *KFTVU Condition.*

Definition 6.3 (KFTVU Condition). An instance satisfies the *KFTVU condition* if for all ordered doubles $a \neq b$ of types and ordered doubles x, y of signals, there **do not exist** outcomes o_a and o_b, allocation function $A : S \to O$, and transfer function $T : S \to \mathbb{R}$ such that the following hold:

$$v_a(o_b) + T(x) - ab > v_a(o_a) + T(y) - aa \tag{1}$$

$$A(y) = o_a \tag{2}$$

$$A(x) = o_b \tag{3}$$

$$(\forall s \in S)\ v_b(o_b) + T(x) - bx \geq v_b(A(s)) + T(s) - bs \tag{4}$$

$$(\forall s \in S)\ v_a(o_a) + T(y) - ay \geq v_a(A(s)) + T(s) - as \tag{5}$$

Intuitively, this condition asks whether we can directly construct a counterexample to the revelation principle. In particular, (1) ensures that a will always misemit to b if we try to implement our choice function truthfully.

Note that if the FTVU condition holds on $a \neq b$, and x, y, then the KFTVU condition also holds on a, b, and x, y. This is because if (1) holds, then (5) does not when $s = x$.

Naïvely, checking this condition requires searching through all allocation functions A and transfer functions T, which would take exponential time. However, the condition can in fact be checked efficiently.

Proposition 6.4. *The KFTVU condition can be checked in polynomial time.*

Proof. For each $a \neq b$ and x, y, we can efficiently check whether the KFTVU condition holds for every o_a, o_b, as follows.

Set $A(y) = o_a$, $A(x) = o_b$, and the rest of A arbitrarily. This satisfies (2) and (3).

Any transfer function that satisfies (1), (4), and (5) also satisfies the following constraints:

$$v_a(o_b) + T(x) - ab > v_a(o_a) + T(y) - aa \qquad \textit{this is condition (1)}$$

$$v_b(o_b) + T(x) - bx \geq v_b(o_a) + T(y) - by \qquad \textit{by condition (4), setting } s = y$$

$$v_a(o_a) + T(y) - ay \geq v_a(o_b) + T(x) - ax \qquad \textit{by condition (5), setting } s = x$$

Furthermore, any transfer function T that satisfies these constraints satisfies (1), and can be transformed into a T' which also satisfies (4) and (5). Let $T' = T$ except for x, y where $T' = T + L$ for some very large L.

Thus a transfer function that satisfies (1), (4), and (5) exists iff one exists that satisfies the constraints. And we can check this using a simple linear feasibility program. □

Theorem 6.5. *The RP holds with known valuations, fixed transfers, and variable utilities iff the* KFTVU condition *holds.*

Proof.
KFTVU condition not holding $\implies$ *RP does not hold*
Let $a \neq b, x, y, o_a, o_b, A : S \to O$, and $T : S \to \mathbb{R}$ be a witness for the violation of the KFTVU condition.

Consider the mechanism N defined by $A_N = A$ and $T_N = T$. By the conditions, there exists an optimal response R_N where $R_N(a) = y$ and $R_N(b) = x$, so that $A_N(R_N(a)) = o_a$ and $A_N(R_N(b)) = o_b$.

But the function $A_H \triangleq A_N \circ R_N$ is not truthfully implementable with $T_H \triangleq T_N \circ R_N$ (these are both taken to be restricted to signals in $range(G)$ here), because a would prefer to misemit to b by (1).

RP not holding $\implies$ *KFTVU condition does not hold*
Let mechanism N with transfers T_N, allocation A_N, and optimal response profile R_N be a witness to the violation of the revelation principle, i.e., the mechanism H with $T_H \triangleq T_N \circ R_N$ and $A_H \triangleq A_N \circ R_N$ is not truthful.

For that to be the case, there must be some types $a \neq b$ and signals x, y such that in H, a strictly prefers emitting b to emitting a, and in N, $R_N(b) = x$ and $R_N(a) = y$.

(a prefers emitting to some b corresponding to a type as WLOG we can assume large negative transfers on signals corresponding to no type in H.)

Now, let $o_a = A_H(a)$, $o_b = A_H(b)$, $A = A_N$, and $T = T_N$. Then, in the KFTVU conditions,
(1) holds because a prefers misemitting b in H;
(2) holds because $o_a = A_H(a) = A_N(R_N(a)) = A(y)$;
(3) holds because $o_b = A_H(b) = A_N(R_N(b)) = A(x)$;
(4) holds because $R_N(b) = x$; and
(5) holds because $R_N(a) = y$.

Hence the KFTVU condition is violated. □

6.4.3 Example: Difference in Revelation Principles for Fixed Transfers and No Transfers in Equilibrium. The following shows that in the type-reporting, known-valuations setting, the 'fixed transfers' and 'no transfers in equilibrium' revelation principles must differ.

Consider an instance with types a, b, c, outcome o, and costs:

$ab = 0$
$bc = 0$

$aa = bb = cc = 0$ (since we are in the type-reporting setting)
and all other costs are ∞.

The only possible choice function for this instance is $F(\cdot) = o$.

The revelation principle does not hold with fixed transfers. We can non-truthfully implement F with a mechanism where b gets a transfer of 10 for reporting c, and a gets 0 at a. But, we cannot truthfully implement F with the same transfers, because a would misemit to b.

On the other hand, with no transfers in equilibrium the revelation principle holds. The outcomes and payments are the same at all signals, so no type would have any incentive to misemit.

6.5 Special Case: Known Valuations, No Transfers In Equilibrium, Variable Utilities

We consider the case where the valuation function is known, no mechanism is allowed to make transfers in equilibrium, and given a non-truthful mechanism, we wish to obtain a truthful mechanism that implements the same choice function but can differ in the utility the agent obtains from it.

Definition 6.6 (KNTEVU Condition). An instance satisfies the *KNTEVU condition* if it satisfies the modified version of the KFTVU conditions where we only consider transfer functions T satisfying the following conditions:

(i) $\forall s$, $T(s) = 0$ or $-L$, where $-L$ is a very negative number such that any type would prefer any outcome and a transfer of zero at any (finite-cost) signal, to receiving a transfer of $-L$ (alongside any outcome at any signal).

(ii) $\forall \theta$, $\exists s$ s.t. $\theta s < \infty$ and $T(s) = 0$.

Restriction (i) ensures that no type receives a transfer of more than 0. Restriction (ii) ensures that each type can receive a transfer of at least 0. Note that these restrictions imply $T(x) = T(y) = 0$.

The KNTEVU condition differs from the KFTVU condition. Hence we must separately prove that the KNTEVU condition can be checked in polynomial time.

Naïvely, checking this would require searching through all allocation functions A and transfer functions with $T(s) \in \{0, -L\}$, which would require exponential time. However, again, the condition can in fact be checked efficiently.

Proposition 6.7. *The KNTEVU condition can be checked in polynomial time.*

Proof. For each $a \neq b$ and x, y, we can efficiently check whether the KNTEVU condition holds for every two outcomes o_a and o_b, as follows.

Since $T(x) = T(y) = 0$, we can directly plug in o_a and o_b into (1) to verify whether it holds.

So now for every o_a and o_b satisfying (1) we need to check that there exists a combination of outcomes and transfers for every $s \neq x, y$ that satisfies (4), (5), (i), and (ii). (And a, y, b, and x continue to be fixed.)

First we check whether there exists a combination of outcomes for every $s \neq x, y$ that satisfies (4) and (5) when $T(s) = 0$. Whether an outcome satisfies these conditions for a single s is independent of which outcome we choose for any other

s'. Hence, all that needs to be checked is, for each s individually, whether there exists an outcome satisfying (4) and (5). If this is the case, we can set $T(\cdot) = 0$. Hence, (ii) holds by default, so we are done.

Otherwise, for each s which has a satisfying outcome, give it that outcome with a transfer of 0. For each s which does not, to satisfy (4) and (5) we have no other option than to give it a transfer of $-L$ (and an arbitrary outcome). So now we simply check whether (ii) still holds. □

Theorem 6.8. *The RP holds with known valuations, no transfers in equilibrium and variable utilities iff the* KNTEVU condition *holds.*

Proof.

KNTEVU condition not holding $\implies$ *the RP does not hold*

Let $a \neq b, x, y, o_a, o_b, A : S \to O$, and $T : S \to \mathbb{R}$ be a witness for the violation of the KNTEVU condition.

Consider the mechanism N defined by $A_N = A$ and $T_N = T$. By the conditions, there exists an optimal response R_N where $R_N(a) = y$ and $R_N(b) = x$, so that $A_N(R_N(a)) = o_a$ and $A_N(R_N(b)) = o_b$. Additionally we have $T_N(R_N(\theta)) = 0$ for all θ by (i) and (ii).

But then the function $A_H \triangleq A_N \circ R_N$ is not truthfully implementable with $T_H \triangleq T_N \circ R_N$ (these are both taken to be restricted to signals in $range(G)$ here), because a would prefer to misemit to b by (1).

RP not holding $\implies$ *KNTEVU condition does not hold*

Let mechanism N with transfers T_N, allocation A_N, no transfers in equilibrium, and optimal response profile R_N be a witness to the violation of the revelation principle. Thus any mechanism H with $A_H \triangleq A_N \circ R_N$ (these are both taken to be restricted to signals in $range(G)$ here), and no transfers in equilibrium is not truthful.

For that to be the case, there must be some types $a \neq b$ and signals x, y such that in H, a strictly prefers emitting b to emitting a, and in N, $R_N(b) = x$ and $R_N(a) = y$.

(a prefers emitting to some b corresponding to a type as WLOG we can assume large negative transfers on signals corresponding to no type in H.)

Create $N' = N$ except that $T_{N'} = -L$ for any signal not emitted by any type under N. Since no type would ever want to receive a transfer of $-L$, $R_{N'} = R_N$. Hence N' has no transfers in equilibrium. And since $A_{N'} = A_N$, we have $A_H = A_{N'} \circ R_{N'}$.

Now, let $o_a = A_H(a)$, $o_b = A_H(b)$, $A = A_{N'}$, and $T = T_{N'}$. Then, in the KNTEVU conditions,
(1) holds because a prefers misemitting b in H;
(2) holds because $o_a = A_H(a) = A_{N'}(R_{N'}(a)) = A(y)$;
(3) holds because $o_b = A_H(b) = A_{N'}(R_{N'}(b)) = A(x)$;
(4) holds because $R_{N'}(b) = x$;
(5) holds because $R_{N'}(a) = y$;
(i) holds by construction of N';
(ii) holds because under N' each type θ can receive a transfer of 0 at $R_N(\theta)$.

Hence the KFTVU condition is violated. □

6.5.1 Example: Difference in Revelation Principles for No Transfers in Equilibrium and No Transfers At All. The following shows that in the type-reporting, known-valuations setting, the 'no transfers in equilibrium' and 'no transfers at all' revelation principles must differ.

Consider an instance with types a, b, c, outcomes: o_1, o_2, valuations:

$v_a(o_1) = 2, v_a(o_2) = 0$
$v_b(o_1) = v_b(o_2) = 1$
$v_c(o_1) = 0, v_c(o_2) = 2,$

and costs:

$ab = cb = 0$
$ba = bc = 1$
$ac = ca = 10$
$\theta\theta = 0$ (since we are in the type-reporting setting).

If we have no transfers *at all*, the revelation principle holds. In any implementation b will always report b, and a and c will never report each other. WLOG assume a non-truthful implementation involves a (but not c) reporting b. We can create a truthful implementation of the same choice function by putting the outcome from b at a as well, and c will still not report a or b.

If we have no transfers *in equilibrium* we are allowed to put negative transfers on types no one will report. Then we can non-truthfully implement $F(a) = o_2,\ F(b) = o_2$, and $F(c) = o_1$ by having a and b report a, c report c, and putting a large negative transfer at b. But we cannot do this truthfully, because if we put o_2 at b then c will report b. Hence the revelation principle does not hold.

6.6 Special Case: Known Valuations, No Transfers At All, Variable Utilities, Type-Reporting Setting

We consider the case where the valuation function is known, we are in a type reporting setting, no mechanism is allowed to make transfers, and given a non-truthful mechanism, we wish to obtain a truthful mechanism that implements the same choice function but can differ in the utility the agent obtains from it.

The condition is essentially the same as that for fixed transfers.

Definition 6.9 (KNTAAVU Condition). An instance satisfies the *KNTAAVU condition* if it satisfies the modified version of the KFTVU condition where we only consider transfer functions T with $T(\cdot) = 0$.

The KNTAAVU condition differs from the KFTVU condition. Hence we must separately prove that the KNTAAVU condition can be checked in polynomial time.

Naïvely, checking this would require searching through all allocation functions A, which would require exponential time. However, again, the condition can in fact be checked efficiently.

Proposition 6.10. *The KNTAAVU condition can be checked in polynomial time.*

Proof. For each $a \neq b$ and x, y, we can efficiently check whether the KNTAAVU condition holds for every two outcomes o_a and o_b, as follows.

We need to check that there exists a combination of outcomes for every $s \neq x, y$ that satisfies (4) and (5). Whether an outcome satisfies these conditions for a single s is independent of which outcome we choose for any other s'. Hence, all that needs to be checked is, for each s individually, whether there exists an outcome satisfying the conditions. □

Theorem 6.11. *The RP holds with known valuations, no transfers at all and variable utilities in the type-reporting setting iff the* KNTAAVU condition *holds.*

Proof. We carry over the proof of Theorem 6.5 unmodified except for requiring $T(\cdot) = 0$ for any transfer function T. The only reason this proof required a transfer function not fixed at zero was to put large negative transfers on unused signals to derive a truthful mechanism. But this is unnecessary in the type-reporting case as every signal will be emitted by some type in the truthful mechanism.

□

6.6.1 Revisiting the Running Example. We now return to the running example for the FTVU case. Recall from 5.3.4 that when the city does not care that the types all receive the same utility, and is unable to make welfare transfers, the revelation principle does not hold for *all* valuation functions, for the cost function under consideration here.

However, it does hold for the *specific* valuation function under consideration here. This is because for the revelation principle to be violated, there would need to be a pair of outcomes such that the difference in valuations between them could incentivize *West* to travel to *East* or *South*, or *South* to *East* or *West*. But there exists no such pair. Thus we only need to search through the space of truthful mechanisms to find an optimal mechanism. The following truthful mechanism obtains the best objective value of 30.

$H =$

	$\hat{North}$	$\hat{West}$	$\hat{East}$	$\hat{South}$
A	*fiber*	*vitamins*	*vitamins*	*vitamins*
T	0	0	0	0

6.6.2 Special Case: No Transfers At All, Signaling Setting. Finding useful conditions that ensure that the RP holds in this case is currently an open problem.

6.7 Known Valuations, Fixed Transfers, Fixed Utilities

We consider the case where the valuation function is known, and given a non-truthful mechanism, we wish to obtain a truthful mechanism that implements the same choice function, makes the same transfers to each agent, and maintains the utility that each type obtains.

Theorem 6.12. *The RP holds with known valuations, fixed transfers, and fixed utilities iff the FTFU condition holds.*

Proof. We can carry over the proof of Theorem 5.13 unmodified. For the *if* direction it showed the RP holds for all possible valuation functions, so will continue to do so for a specific one. For the *only if* direction we used a constant choice function whose choice (and hence valuation) of outcome did not matter. □

7 RESULTS: REVELATION PRINCIPLE FOR FULLY SPECIFIED INSTANCES

In the previous section, we considered the case where we know the valuation function, and wish to know if the revelation principle holds for it. All that is needed to violate the revelation principle is for there to exist *some* choice function (possibly together with transfers and/or utilities) that can be non-truthfully, but not truthfully, implemented. But this may be of little interest if we already know the precise choice function (etc.) we wish to implement.

Indeed, even if the revelation principle does not hold for all choice functions (etc.), it may yet hold for the one we care about. We study this here. Hence, an instance is now a *fully specified instance*, consisting of Θ, S, c, O, and v as before, but also F, possibly a specific transfer function $T^* : \Theta \rightarrow \mathbb{R}$, and possibly a specific utility function $U^* : \Theta \rightarrow \mathbb{R}$, which we wish to implement.

Definition 7.1 (Revelation Principle on a Fully Specified Instance). We say the RP is true for a fully specified instance if either (1) a truthful mechanism T exists that implements the choice function (with the required utilities and/or transfers), or (2) no (possibly non-truthful) mechanism N does.

We will show that deciding whether the RP holds on individual fully specified instances (even in the type-reporting setting) reduces to the computational problem of deciding whether a (possibly non-truthful) implementation exists.

Lemma 7.2. *Determining whether the revelation principle fails to hold on a given fully specified instance is computationally exactly as hard as determining whether there is a (not necessarily truthful) implementation for that instance.*

Proof. In each case, we can efficiently verify whether there is a truthful implementation for that instance:

- If there are no transfers, there is only one possibility for what the mechanism does for the signals in $range(G)$. But, we must also assign outcomes to the signals outside $range(G)$ such that no type misemits to them. Whether there exists such an outcome for a given s outside $range(G)$ is independent of which outcome we choose for any other such s'. Hence, all that needs to be checked is, for each s individually, whether there exists an outcome such that no type will misemit to it.
- If there are transfers but they are fixed (or implicitly fixed because utilities are), again there is only one possibility for what the mechanism does for signals in $range(G)$. In this case, we can always ensure that no type will misemit outside $range(G)$, by putting a sufficiently negative transfer on those signals.
- Finally, if neither transfers nor utilities are fixed, then it is a simple linear feasibility problem to determine whether transfers exist that implement the choice function on the signals in $range(G)$. And again, we can ensure that no type misemits to signals outside $range(G)$ by putting sufficiently negative transfers there.

Hence, we can reduce the problem of determining whether a (not necessarily truthful) implementation exists for a fully specified instance to the problem of checking whether the RP holds on a fully specified instance, as follows. First, check

whether a truthful implementation exists; if so the answer is "yes." Otherwise, there is an implementation if and only if the revelation principle fails to hold on this instance.

Conversely, we can reduce the problem of checking whether the RP holds on a fully specified instance to the problem of determining whether a (not necessarily truthful) implementation exists for a fully specified instance, as follows. Again, first, check whether a truthful implementation exists; if so the answer is "yes." Otherwise, the revelation principle holds if and only if there is no implementation. □

Theorem 7.3. *Computing whether the revelation principle holds on a given fully specified instance is coNP-complete (whether or not transfers and/or utilities are fixed). This is true even in the type-reporting setting.*

Proof. *Variable Transfers*
When we have variable transfers, for both variable and fixed utilities, the problem of determining whether a (not necessarily truthful) implementation for an instance exists is NP-complete [1, 15].

Fixed Transfers, Variable Utilities
[1] showed that implementation is NP-complete in the type-reporting setting with no transfers at all and variable utilities. In particular, for an arbitrary 3-SAT instance they show how to construct a mechanism design with partial verification instance with choice function F, such that the 3-SAT instance is satisfiable if and only if F is implementable with no transfers at all.

This does not necessarily imply that the case with no transfers *in equilibrium* is NP-complete. These cases may differ, as it is possible that non-truthful implementation requires giving negative transfers to types not reported.

But, as it turns out, the proof that they give can also show that implementation with no transfers in equilibrium is NP-complete. On their instance, F is implementable with no transfers at all *if and only if* it is implementable with no transfers in equilibrium. The 'only if' is automatic as no transfers at all is a special case of no transfers in equilibrium. For the 'if' to hold, we must show that allowing negative transfers to types not reported cannot cause F to become implementable. This is equivalent to asking if forbidding certain types from being reported is helpful. For the instances they construct, it is not. The fundamental difficulty in implementation in their instances stems from reconciling the following two needs:

- Every clause type needs to receive an outcome of 'true' at some literal type in the clause.
- Every variable type needs to receive an outcome of 'false' at one of its two literal types.

Forbidding certain type reports will not help satisfy either of these conditions. Hence, implementability with no transfers in equilibrium and variable utilities is NP-complete, and thus fixed transfers to the agent and variable utilities is as well.

Fixed Transfers, Fixed Utilities
Finally, [15] showed that in the partial verification case, when we have no transfers in equilibrium, the NP-completeness of the variable utilities case implies the NP-completeness of the fixed utilities case. Hence implementability with no transfers in equilibrium with fixed utilities is NP-complete, and thus implementability with fixed transfers to the agent is as well.

So, the problem of determining whether a (not necessarily truthful) implementation for an instance exists is NP-complete in all the cases. Hence, Lemma 7.2 implies that determining whether the revelation principle **fails** to hold is NP-complete, proving the theorem. □

ACKNOWLEDGMENTS

We are thankful for support from ARO under grants W911NF-12-1-0550 and W911NF-11-1-0332, NSF under awards IIS-1527434, IIS-0953756, IIS-1814056, CCF-1101659, and CCF-1337215, and a Guggenheim Fellowship. This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing.

REFERENCES

[1] Vincenzo Auletta, Paolo Penna, Giuseppe Persiano, and Carmine Ventre. 2011. Alternatives to truthfulness are hard to recognize. *Autonomous Agents and Multi-Agent Systems* 22, 1 (2011), 200–216.
[2] Marco Barreno, Blaine Nelson, Anthony D. Joseph, and J. D. Tygar. 2010. The security of machine learning. *Machine Learning* 81, 2 (2010), 121–148.
[3] Jesse Bull and Joel Watson. 2004. Evidence disclosure and verifiability. *Journal of Economic Theory* 118, 1 (2004), 1–31.
[4] Jesse Bull and Joel Watson. 2007. Hard evidence and mechanism design. *Games and Economic Behavior* 58, 1 (2007), 75–93.
[5] Ioannis Caragiannis, Edith Elkind, Mario Szegedy, and Lan Yu. 2012. Mechanism design: from partial to probabilistic verification. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*. Valencia, Spain, 266–283.
[6] Vincent Conitzer and Tuomas Sandholm. 2002. Complexity of Mechanism Design. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. Edmonton, Canada, 103–110.
[7] Vincent Conitzer and Tuomas Sandholm. 2004. Self-interested Automated Mechanism Design and Implications for Optimal Combinatorial Auctions. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*. New York, NY, USA, 132–141.
[8] Nilesh N. Dalvi, Pedro Domingos, Mausam, Sumit K. Sanghai, and Deepak Verma. 2004. Adversarial Classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Seattle, WA, USA, 99–108.
[9] Raymond Deneckere and Sergei Severinov. 2008. Mechanism design with partial state verifiability. *Games and Economic Behavior* 64, 2 (2008), 487–513.
[10] Raymond Deneckere and Sergei Severinov. 2014. Optimal Screening with Costly Misrepresentation. (2014). http://www.severinov.com/working_papers/screen_costly_nov2014.pdf Working Paper.
[11] Jerry Green and Jean-Jacques Laffont. 1986. Partially verifiable information and mechanism design. *Review of Economic Studies* 53 (1986), 447–456.
[12] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. 2016. Strategic Classification. In *Innovations in Theoretical Computer Science (ITCS)*. Cambridge, MA, USA.
[13] Navin Kartik and Olivier Tercieux. 2012. Implementation with evidence. *Theoretical Economics* 7, 2 (2012), 323–355.
[14] Navin Kartik, Olivier Tercieux, and Richard Holden. 2014. Simple mechanisms and preferences for honesty. *Games and Economic Behavior* 83 (2014), 284–290.
[15] Andrew Kephart and Vincent Conitzer. 2015. Complexity of Mechanism Design with Signaling Costs. In *Proceedings of the Fourteenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. Istanbul, Turkey, 357–365.
[16] Frederic Koessler and Eduardo Perez-Richet. 2014. *Evidence based mechanisms*. Technical Report. working paper. http://www.tse-fr.eu/sites/default/files/TSE/documents/sem2015/eco_theo/perez_richet.pdf
[17] Jeffrey Lacker and John Weinberg. 1989. Optimal Contracts under Costly State Falsification. *Journal of Political Economy* 97, 6 (1989), 1345–63.
[18] Jean-Charles Rochet. 1987. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics* 16, 2 (April 1987), 191–200.
[19] Michael Spence. 1973. Job Market Signaling. *Quarterly Journal of Economics* 87, 3 (1973), 355–374.
[20] Roland Strausz. 2016. Mechanism Design with Partially Verifiable Information. 2040 (2016).
[21] Lan Yu. 2011. Mechanism design with partial verification and revelation principle. *Autonomous Agents and Multi-Agent Systems* 22, 1 (2011), 217–223.