Probabilistic Verification for Obviously Strategyproof Mechanisms

Diodato Ferraioli
University of Salerno, Italy
dferraioli@unisa.it

Carmine Ventre
University of Essex, UK
c.ventre@essex.ac.uk

Abstract

Obviously strategyproof (OSP) mechanisms maintain the incentive compatibility of agents that are not fully rational. They have been object of a number of studies since their recent definition. A research agenda, initiated in [Ferraioli and Ventre, 2017], is to find a small set (possibly, the smallest) of conditions allowing to implement an OSP mechanism. To this aim, we define a model of probabilistic verification wherein agents are caught misbehaving with a certain probability, and show how OSP mechanisms can implement every social choice function at the cost of either imposing very large fines or verifying a linear number of agents.

1 Introduction

Will people strategize against an incentive-compatible mechanism? The answer depends on whether they will *understand* that doing so is against their own interest and, ultimately, on their rationality and cognitive skills. This question has often been raised in literature (see, e.g., [Sandholm and Gilpin, 2003; Ferraioli *et al.*, 2015]) and much of the recent research in mechanism design is motivated by this question. Several definitions for "simple" mechanisms have been recently given in literature [Hartline and Roughgarden, 2009; Chawla *et al.*, 2010; Babaioff *et al.*, 2014; Adamczyk *et al.*, 2015; Brânzei and Procaccia, 2015]. This quest for the right definition for simple mechanisms culminated with the introduction of *obviously strategyproof* (OSP) mechanisms [Li, 2017].

Obvious strategyproofness focuses on how a mechanism is executed (e.g., English auction vs. sealed bid second price auction), and requires that whenever an agent takes an action during the execution of the mechanism, the "truthful behavior" must be dominant for that agent, even if no reasoning is done about the future actions of other agents. This concept is motivated by the experimental evidence that some mechanism implementations (e.g., clock auction) are easier to understand than other theoretically equivalent ones (e.g., sealed-bid auction). OSP mechanisms have also solid theoretical foundations: they are the only ones that preserve the incentive-compatibility of agents who lack contingent reasoning skills [Li, 2017] and they satisfy a natural generalization of standard decision theory axioms [Zhang and Levin, 2017].

This concept has attracted a considerable amount of recent work [Ashlagi and Gonczarowski, 2015; Bade and Gonczarowski, 2017; Ferraioli and Ventre, 2017; Pycia and Troyan, 2016] that mainly focuses on the properties and the limitations of these mechanisms. Of particular interest for our study are the results proved by [Ferraioli and Ventre, 2017] showing that OSP mechanisms cannot have good approximation guarantees for machine scheduling and facility location, two canonical optimization problems studied in the area. However, monetary transfers are sufficient for the existence of optimal OSP mechanisms when the designer can "monitor" all agents (meaning that a lying agent is artificially made to receive an utility that is the worst between the one computed according to her true type and the one computed as if her type coincided with her bid). Since money is often undesirable (cf. [Procaccia and Tennenholtz, 2013]) our main aim here is to understand whether there are novel ways to exert control over the agents that can reconcile approximation and OSP mechanisms with limited or absent transfer of money.

Our Contribution. We introduce a model of *probabilistic verification* wherein the mechanism designer has access to a (potentially faulty) verification device that she can use at runtime to check whether an agent has lied or not. The device will catch the lie of the checked agent with certainty, or with a certain probability if faulty. For example, whenever the type t of an agent is her location on the real line (as in facility location), the designer can use a GPS logger to check whether the agent location is the same as her reported type b. In our terminology, such a tool is faulty if its reading t' of t is subject to some measurement error δ and, therefore, the agent would be caught only if $|b - t'| > \delta$; more generally, one could imagine different tools that make mistakes in their measurements with some probability rather than in range (e.g., it gets better as the difference between reported and real type increases). This notion generalizes and combines the different notions of verification introduced in related literature (see, e.g., [Caragiannis *et al.*, 2012; Penna and Ventre, 2014]). With respect to monitoring, the mechanism designer has in our probabilistic model a more general way to define fines for lying agents, whilst, on the other hand, might have a faulty verification device.

We begin by studying what we call the *full probabilistic verification model*, wherein every agent is verifiable. We

prove that, in this setting, it is possible to obtain an OSP mechanism for every specific problem of interest; we essentially show that we can always define verification probabilities and fines to make any lie obviously dominated. On the technical level, we show that there is a trade-off between the kind of verification device needed (i.e., the verification probabilities) and the amount of fines imposed to lying agents that are caught. Our results imply that we can set the fines so that only a constant number of agents is verified in expectation.

The result above requires all the agents to be verifiable – in some contexts this might be impossible (e.g., not all the agents might have been equipped with a GPS logger) or expensive (e.g., in tax auditing). We therefore look at the *partial probabilistic verification model*, where for some agents we cannot use any verification (and so the combination of fines and probabilistic checks will not make lying obviously dominated). In the main technical contribution of this work, we prove that there is a problem such that all ε-OSP mechanisms (i.e., agents will not deviate for small gains ε) that solve this problem need to verify in expectation a linear number of agents. We focus on the well studied *public project problem* [Jackson and Moulin, 1992] and identify a small domain for which "many" agents need to be verified by any ε-OSP mechanism that solves the problem.

We finally prove that this result is tight for mechanisms that implement a social choice function only asymptotically and, in expectation, are ε-OSP. In detail, we connect OSP with *differential privacy* and show how the exponential mechanism [Nissim *et al.*, 2012] can be implemented with partial probabilistic verification, so that, in expectation, it becomes ε-OSP and verifies $n - o(n)$ agents. Although the proofs basically follow the known ones, we regard this result interesting for two main reasons (in addition to showing the tightness of the lower bound). Firstly, it shows how, through verification, differential privacy can be related to OSP just like truthfulness. Secondly, the mechanism becomes implementable not just in our probabilistic framework but also with selective verification (cf, e.g., [Fotakis *et al.*, 2016]).

2 Preliminaries

A mechanism design setting is defined by a set of n *selfish agents* and a set of allowed *outcomes* $\mathcal{S}$. Each agent i has a *type* $t_i \in D_i$, where D_i is called the *domain* of i. The type t_i is assumed to be *private knowledge* of agent i. Each selfish agent i has a *valuation function* $v_i \colon D_i \times \mathcal{S} \to \mathbb{R}$. For $t_i \in D_i$ and $X \in \mathcal{S}$, $v_i(t_i, X)$ is the valuation that agent i has for outcome X when her type is t_i. We will often use $t_i(X)$ as a shorthand for $v_i(t_i, X)$. The domain D_i of agent i is *bounded* if $t_i(X) \in [t_{\inf}, t_{\sup}]$ for all i, $t \in D_i$, $X \in \mathcal{S}$.

A *mechanism* is a process for selecting an outcome $X \in \mathcal{S}$. To this aim, the mechanism interacts with agents. Specifically, agent i is observed to take *actions* (e.g., saying yes/no) that may depend on her presumed type $b_i \in D_i$ (e.g., saying yes could "signal" that the presumed type has some properties that b_i alone might enjoy). We say that agent i takes *actions according to* b_i to stress this. Observe that the presumed type b_i can be different from the real type t_i. For a mechanism $\mathcal{M}$, we let $\mathcal{M}(\mathbf{b})$ denote the outcome returned when agents take actions according to types $\mathbf{b} = (b_1, \ldots, b_n)$.

A mechanism $\mathcal{M}$ is *strategy-proof* if for every i, every $\mathbf{b}_{-i} = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$ and every $b_i \in D_i$, $v_i(t_i, \mathcal{M}(t_i, \mathbf{b}_{-i})) \geq v_i(t_i, \mathcal{M}(b_i, \mathbf{b}_{-i}))$, where t_i is the true type of i. That is, in a strategy-proof mechanism it is dominant for all agents to take actions according to the true type.

Given a *social choice function* $f \colon D \to \mathcal{S}$, where $D = D_1 \times \cdots \times D_n$ is the set of type profiles $(b_1, \ldots, b_n)$, a mechanism $\mathcal{M}$ is said to *implement* f (in expectation) if $\mathcal{M}(\mathbf{b}) = f(\mathbf{b})$ ($E[\mathcal{M}(\mathbf{b})] = f(\mathbf{b})$) for every $\mathbf{b}$. $\mathcal{M}$ is instead said to *implement* f *asymptotically* (in expectation) if $\lim_{n\to\infty} \mathcal{M}(\mathbf{b}) = f(\mathbf{b})$ ($\lim_{n\to\infty} E[\mathcal{M}(\mathbf{b})] = f(\mathbf{b})$).

Obvious Strategyproofness. We now define the concept of OSP mechanism. We follow [Ashlagi and Gonczarowski, 2015] and assume w.l.o.g. complete information, but we allow concurrent updates. An *extensive-form mechanism* $\mathcal{M}$ is defined by a directed tree $\mathcal{T} = (V, E)$ such that:

- every leaf ℓ of the tree is labeled by a possible outcome $X(\ell) \in \mathcal{S}$ of the mechanism;
- every internal vertex $u \in V$ either is labeled by an agent $S(u) \in [n]$, or is a *chance vertex* labeled by character c;
- every edge $e = (u, v) \in E$ going out from a non-chance vertex is labeled by a set $T(e) \subseteq D$ of type profiles s.t.:
  - the sets of profiles that label the edges outgoing from the same vertex u are disjoint, i.e., for every triple of vertices u, v, v' such that $(u, v) \in E$ and $(u, v') \in E$, we have that $T(u, v) \cap T(u, v') = \emptyset$;
  - the union of the sets of profiles labeling the edges outgoing from non-root vertex u is equal to the set of profiles labeling the edge going in u, i.e., $\bigcup_{v\colon (u,v)\in E} T(u, v) = T(\phi(u), u)$, where $\phi(u)$ is the parent of u in $\mathcal{T}$;
  - the union of the sets of profiles that label the edges outgoing from the root vertex r is equal to the set of all profiles, i.e., $\bigcup_{v\colon (r,v)\in E} T(r, v) = D$;
  - for every u, v such that $(u, v) \in E$ and for every two profiles $\mathbf{b}, \mathbf{b}' \in T(\phi(u), u)$ such that $b_{S(u)} = b'_{S(u)}$, if $\mathbf{b} \in T(u, v)$, then also $\mathbf{b}' \in T(u, v)$;
- every $e = (u, v) \in E$, u being a chance vertex, has label $T(e) = D$ if u is a root, and $T(e) = T(\phi(u), u)$ otherwise;
- every non-chance vertex u is associated to an information set $I(u) \subseteq D$, where $I(r) = D$, and, for $u \neq r$, either $I(u) = D$ or $I(u) = T(\phi(v), v)$ for some v being a non-root vertex in the path from r to u.

Roughly speaking, the tree represents the steps of the execution of the mechanism. As long as the current visited vertex u is not a leaf, if u is a chance vertex, then the mechanism decides the next step by using its own random coin tosses, otherwise it interacts with the agent in $S(u)$. If the current visited vertex u is a leaf, then it returns the outcome that labels u. Different edges outgoing from a non-chance vertex u are used for modeling the different actions that agents can take during this interaction with the mechanism. In particular, each possible action is assigned to an edge outgoing from u. As suggested above, the action that agent i takes may depend on her presumed type $b_i \in D_i$. That is, different presumed types may correspond to taking different actions, and thus to

different edges. The label $T(e)$ on edge $e = (u, v)$ then lists the type profiles that enable agent $S(u)$ to take those actions that have been assigned to e. In other words, when the agents take the actions assigned to edge e, then the mechanism (and the other agents) can infer that the type profile must be contained in $T(e)$. Clearly, chance vertices do not change the current information available to the mechanism, and thus they do not change the edges' label. The constraints on the edges' label can be then explained as follows: first we can safely assume that different actions must correspond to different type profiles (indeed, if two different actions are enabled by the same profiles we can consider them as a single action); second, we can safely assume that each action must correspond to at least one type profile that has not been excluded yet by actions taken before node u was visited (otherwise, we could have excluded this type profile earlier); third, we have that the action taken by agent $S(u)$ can only inform about types of this agent and not about the type of the other agents. The execution ends when we reach a leaf ℓ of the tree. In this case, the mechanism returns the outcome that labels ℓ.

We do not necessarily assume that agents take their actions sequentially. Indeed, we use information sets to model a mechanism that concurrently interacts with multiple agents. The information set at node u is, indeed, the set of profiles that player $S(u)$ selected at this node believes can be realized given her current information: thus, if i interacts with the mechanism at the same time as agent j, then the set of profiles that can be realized for i is the same as for j. Hence, even if we model i and j as taking their actions at two different nodes, the information sets of these two nodes are exactly the same (i.e., j ignores i's action and viceversa).

Observe that for every profile $\mathbf{b}$ there is only one leaf $\ell = \ell(\mathbf{b})$ such that $\mathbf{b}$ belongs to $T(\phi(\ell), \ell)$. For this reason we say that $\mathcal{M}(\mathbf{b}) = X(\ell)$. Moreover, for every type profile $\mathbf{b}$ and every node $u \in V$, we say that $\mathbf{b}$ is *compatible* with u if $\mathbf{b} \in I(u)$. Finally, two profiles $\mathbf{b}, \mathbf{b}'$ are said to *diverge* at vertex u if there are two vertices v, v' such that $(u, v) \in E$, $(u, v') \in E$ and $\mathbf{b} \in T(u, v)$, whereas $\mathbf{b}' \in T(u, v')$.

Now we define obvious strategyproofness. An extensive-form mechanism $\mathcal{M}$ is *ε-obviously strategy-proof (ε-OSP)* if for every agent i with real type t_i, for every vertex u such that $i = S(u)$, for every $\mathbf{b}_{-i}, \mathbf{b}'_{-i}$ (with $\mathbf{b}'_{-i}$ not necessarily different from $\mathbf{b}_{-i}$), and for every $b_i \in D_i$, with $b_i \neq t_i$, such that $(t_i, \mathbf{b}_{-i})$ and $(b_i, \mathbf{b}'_{-i})$ are compatible with u, but diverge at u, it holds that $v_i(t_i, \mathcal{M}(t_i, \mathbf{b}_{-i})) \geq v_i(t_i, \mathcal{M}(b_i, \mathbf{b}'_{-i})) - \varepsilon$. $\mathcal{M}$ is obviously strategy proof (OSP) if $\varepsilon = 0$. Roughly speaking, an obvious strategy-proof mechanism requires that, whenever agent i is asked to take a decision that depends on her type, the worst valuation that she can get if at this time step she behaves according to her true type is at most the same as the best valuation achievable by behaving as she had a different type.

Observe that if a mechanism is obviously strategy-proof, then it is also strategy-proof. Indeed, the latter requires that, for every $\mathbf{b}_{-i}$, truthtelling is the best choice of agent i. Instead, the definition of OSP requires that truthtelling is the best choice even if all the other players also change their strategy (in a way that is compatible with the action that they previously took during the execution of the mechanism).

Probabilistic Verification. We introduce a general model of probabilistic verification, inspired by [Caragiannis *et al.*, 2012; Penna and Ventre, 2014]. Fix i and $\mathbf{b}_{-i}$. Let t and t' denote the true and reported type of agent i, respectively. A *mechanism with probabilistic verification* $\mathcal{M}$ catches agent i lying with probability $(1 - p^i_{t',t}(\mathbf{b}_{-i}))$ and punishes the agent caught lying with a fine $F^i_{t',t}(\mathbf{b}_{-i}) > 0$. (So $p^i_{t',t}(\mathbf{b}_{-i})$ denotes the probability that the verification has *not* worked – clearly, $p^i_{t,t}(\mathbf{b}_{-i}) = 0$.) We drop i from the notation when clear from the context. We follow the literature and assume that verification occurs after the outcome has been computed. Moreover, when a mechanism with probabilistic verification $\mathcal{M}$ catches agent i lying it acquires knowledge of $t(\mathcal{M}(t', \mathbf{b}_{-i}))$. Except for the fines, the mechanism does not use any other form of transfers. In this sense, our research extends the literature on mechanisms that trade money with verification to ensure incentive-compatibility[1], see, e.g., [Fotakis *et al.*, 2017; Ferraioli *et al.*, 2016]. When misreporting her type to a mechanism with probabilistic verification, agent i will then have a valuation $t(\mathcal{M}(t', \mathbf{b}_{-i})) - (1 - p^i_{t',t}(\mathbf{b}_{-i}))F^i_{t',t}(\mathbf{b}_{-i})$. There are two complementary interpretations of this formula, depending on the power of the verification device used. The first assumes that the verification device is faulty (e.g., subject to measurement errors) and even if i is verified there is a chance that depends on type, bid and what the others reported that she is not caught (e.g., error might depend on the "distance" between t and t'). The second, instead, is closer to the selective verification of [Fotakis *et al.*, 2016] in that the device is faultless and once an agent is selected to be verified she will be fined with certainty if she lied. Naturally, as the mechanism has no knowledge of t, the probability with which the mechanism selects agent i for verification can only depend, in this case, on her identity, report and bids of the others but not on her type t; i.e., $p^i_{t',t}(\mathbf{b}_{-i})$ reads $p^i_{t'}(\mathbf{b}_{-i})$.

We will consider two different categories of mechanisms with probabilistic verification: the full model wherein all the agents are verifiable, so that we can define $p^i_{t',t}(\mathbf{b}_{-i}) \in [0, 1]$ for every $(i, t, t', \mathbf{b}_{-i})$, and the partial model wherein there exists at least one agent i that is not verifiable, i.e., for which $p^i_{t',t}(\mathbf{b}_{-i}) = 1$ for every $\mathbf{b}_{-i}$ and every t, t' with $t \neq t'$. The non-verifiable agents might be given a priori (e.g., agents without GPS loggers) or be determined by the designer's limited resources (e.g., with k out of n loggers available to allocate, there would remain $n - k$ non-verifiable agents).

3 Full Probabilistic Verification

In this section we prove that full probabilistic verification is very powerful. Specifically, we prove the following theorem.

Theorem 1. *If the domains of agents are bounded, then for every social choice function f there is an OSP mechanism with full probabilistic verification that implements f and verifies in expectation only a constant number of agents.*

[1]Indeed, in classical mechanisms money must be effectively transferred to enforce (obvious) strategy-proofness, here fines are, in a sense, only threats and they are never effectively transferred.

Hence, there are social choice functions for which an OSP mechanism is implementable in the full probabilistic verification model but not implementable in the standard model with money (e.g., facility location [Ferraioli and Ventre, 2017]).

Unfortunately, the mechanism of Theorem 1 needs very large fines. However, we prove that full probabilistic verification still turns out to be a powerful tool even if large fines are not available. In particular, we observe that there is a trade-off between fines and the number of verified agents. Hence, one may be able to work with lower fines, by simply having more accurate verification (in a sense, we can reduce fines only if we spend more for our verification tools). Hence, we can show that if an upper bound on fines is given, it is possible, under opportune conditions, to compute verification probabilities such that the resulting mechanism with full probabilistic verification is OSP. We also consider the opposite direction. That is, we assume that verification probabilities are given, and investigate the lowest fines that one needs to set in order to have an OSP mechanism. Hence, our results actually consider all the following possible cases: (i) we are given bounded fines, and we need to decide how faulty we can allow the verification device to be in order to have an OSP mechanism (cf. Lemma 2); (ii) the faultiness of the verification device is given, and we need to set fines for the mechanism to be OSP (cf. Proposition 3); (iii) we require that the expected number of verified agents is limited, and we design a corresponding OSP mechanism, by computing both the faultiness of the verification device and the fines (cf. Theorem 1).

OSP Mechanism with Few Verified Agents. For every i, let $F^i_{\arg} = \arg_{t,t',\mathbf{b}_{-i}} F^i_{t',t}(\mathbf{b}_{-i})$, with $\arg \in \{\min, \max\}$.

Lemma 2. *For every social choice function f and fines $F^i_{t',t}(\mathbf{b}_{-i})$ such that $F^i_{\min} \geq t_{\sup} - t_{\inf}$ for all i, let $\mathcal{M}_F$ be the mechanism with full probabilistic verification that requires agents to sequentially reveal their type, implements f, uses fines $F^i_{t',t}(\mathbf{b}_{-i})$, and sets $p^i_{t',t}(\mathbf{b}_{-i}) = \frac{t_{\inf}-t_{\sup}+F^i_{\min}}{F^i_{\max}}$. Then $\mathcal{M}_F$ is OSP.*

Proof. The mechanism is well defined since $F^i_{\min} \geq t_{\sup} - t_{\inf}$ implies $p^i_{t',t}(\mathbf{b}_{-i}) \geq 0$ for every $(i,t,t',\mathbf{b}_{-i})$. Moreover, since $t_{\inf} < t_{\sup}$ and $F^i_{\min} \leq F^i_{\max}$ then we get $p^i_{t',t}(\mathbf{b}_{-i}) < 1$ for every $(i,t,t',\mathbf{b}_{-i})$.

For OSP-ness, fix agent i. Since $p^i_{t',t}(\mathbf{b}'_{-i}) < 1$ for each $(t,t',\mathbf{b}'_{-i})$, then $1 - p_{t',t}(\mathbf{b}_{-i}) = 1 - \frac{t_{\inf}-t_{\sup}+F^i_{\min}}{F^i_{\max}} = \frac{(F^i_{\max}-F^i_{\min})+(t_{\sup}-t_{\inf})}{(F^i_{\max}-F^i_{\min})+F^i_{\min}} \geq \frac{t_{\sup}-t_{\inf}}{F^i_{\min}} \geq \frac{t_{\sup}-t_{\inf}}{F^i_{t',t}(\mathbf{b}_{-i})}$, where the first inequality follows since, for every $C \geq 0$, $\frac{C+x}{C+y} \geq \frac{x}{y}$ whenever $x \leq y$. The lemma follows since, for every $\mathbf{b}_{-i}$, $t(\mathcal{M}(t,\mathbf{b}_{-i})) \geq t_{\inf} = t_{\sup} - (1-p_{t',t}(\mathbf{b}_{-i}))\frac{t_{\sup}-t_{\inf}}{1-p_{t',t}(\mathbf{b}_{-i})} \geq t_{\sup} - (1-p_{t',t}(\mathbf{b}_{-i}))F^i_{t',t}(\mathbf{b}_{-i}) \geq t(\mathcal{M}_F(t',\mathbf{b}'_{-i})) - (1-p_{t',t}(\mathbf{b}'_{-i}))F^i_{t',t}(\mathbf{b}'_{-i})$.

Lemma 2 allows us to understand how the choice of $F^i_{\min}$ and $F^i_{\max}$ changes the probabilities. In particular, $p^i_{t',t}$ is higher when $F^i_{\min} = F^i_{\max} = F$. Moreover, this probability quickly grows to 1 as F increases (simply look at the derivative of $(-t_{\sup} + t_{\inf} + x)/x$): this shows that according to the choice of fines, there is a sort of all-or-nothing verification, in which one quickly passes from mechanisms requiring very precise verification devices to mechanisms that can be implemented even in presence of very faulty devices.

Proof of Theorem 1. Set, for each $(i,t,t',\mathbf{b}_{-i})$, $F^i_{t',t}(\mathbf{b}_{-i}) = \gamma(t_{\sup} - t_{\inf})$ for some $\gamma > 1$ that will be fixed later, and let $\mathcal{M}_F$ be the mechanism with full probabilistic verification that requires agents to sequentially reveal their type, implements f, sets $p^i_{t',t}(\mathbf{b}_{-i}) = \min\left\{1, \frac{t_{\inf}-t_{\sup}+F^i_{\min}}{F^i_{\max}}\right\}$, and uses fines $F^i_{t',t}(\mathbf{b}_{-i})$. Since $F^i_{\min} = F^i_{\max} = F^i_{t',t}(\mathbf{b}_{-i}) \geq t_{\sup} - t_{\inf}$, then, according to Lemma 2, $\mathcal{M}_F$ is OSP. It verifies $\sum_{i=1}^n (1-p^i_{b_i,t_i}(\mathbf{b}_{-i})) = n - \sum_{i=1}^n \left(1-\frac{1}{\gamma}\right) = \frac{n}{\gamma}$ agents in expectation. The theorem follows by taking $\gamma = \Omega(n)$. □

OSP-ness for Given Verification Probabilities. Fix i and let t be her true type. Let $p^i_{\max} = \max_{\mathbf{b}_{-i}, b \in D_i} p_{b,t}(\mathbf{b}_{-i})$.

Proposition 3. *For every social choice function f and verification probabilities p such that $p^i_{t',t}(\mathbf{b}_{-i}) \neq 1$ for all $(i,\mathbf{b}_{-i},t,t')$, let $\mathcal{M}_p$ be the mechanism with full probabilistic verification that requires agents to sequentially reveal their type, implements f, uses fines $F^i_{t',t}(\mathbf{b}_{-i}) \geq \frac{t(\mathcal{M}(t',\mathbf{b}_{-i}))-t_{\inf}}{1-p^i_{\max}}$, and verification probabilities p. Then $\mathcal{M}_p$ is OSP.*

Proof. Since $p^i_{\max} \neq 1$, then

$$\inf_{\mathbf{b}_{-i}} t(\mathcal{M}_p(t,\mathbf{b}_{-i})) \geq t_{\inf} \geq \sup_{\mathbf{b}_{-i}} \Big\{ t(\mathcal{M}_p(t',\mathbf{b}_{-i})) - (1-p_{t',t}(\mathbf{b}_{-i}))\frac{t(\mathcal{M}_p(t',\mathbf{b}_{-i})) - t_{\inf}}{1-p^i_{\max}} \Big\}$$
$$\geq \sup_{\mathbf{b}_{-i}} \left\{ t(\mathcal{M}_p(t',\mathbf{b}_{-i})) - (1-p_{t',t}(\mathbf{b}_{-i}))F_{t',t}(\mathbf{b}_{-i}) \right\}. \quad \square$$

A close inspection to the proof reveals that our lower bounds on fines are tight, i.e., for any t, t', i and $\mathbf{b}_{-i}$ there is no smaller $F^i_{t',t}(\mathbf{b}_{-i})$ that would guarantee OSP. This in particular means that once the probabilities to verify have been set there is not much flexibility in the fines imposed on agents.

4 Partial Probabilistic Verification

One limitation of Theorem 1 is that the fines have a very high value (linear in the number of agents) which makes their enforceability doubtful. Looking at the proof, if the mechanism were able to verify a constant number of agents, then the summation would have a constant number of addends, γ would become a constant and, ultimately, the fines would be reduced significantly. Here, we investigate whether it is possible to obtain OSP mechanisms that verify few agents.

We let $V(\mathbf{b})$ denote the subset of verifiable agents for type profile $\mathbf{b}$. Such a subset, just like the outcome, can be chosen randomly and can depend on agents' declaration. Note that $V(\mathbf{b}) = n$ in the full probabilistic verification model. Here, we consider partial probabilistic verification, and we ask how large should $V(\mathbf{b})$ be in order to have an OSP mechanism.

As in Proposition 3, for bounded domains we can guarantee through fines that, no matter the quality of the verification device, truthtelling will be obviously dominant for all the agents in $V(\mathbf{b})$. Therefore, the mechanism "only" needs to obviously incentivize the agents that are not in $V(\mathbf{b})$. Interestingly, we next prove the number of these agents needs to be small for any OSP mechanism with partial probabilistic verification, even in the case in which the designer can choose the agents to verify in $V(\mathbf{b})$. In fact, we next show that for every $\varepsilon > 0$ there is a problem for which every ε-OSP mechanism needs to verify at least $n - o(n)$ agents, where n is the total number of agents. We prove that this bound is tight: for every $\varepsilon > 0$ there is a mechanism that, in expectation, is ε-OSP, and it is able to implement asymptotically every social choice function, by verifying at most $n - o(n)$ agents.

How Many Agents Must Be Verified? Consider the following problem, known as *public project* problem. We need to decide whether to implement or not a public project (e.g., building a bridge) whose cost is c. The society is comprised of a set N of n individuals (also termed agents or customers) that we denote as integers from 0 to $n-1$. The valuation of agent i if the project is implemented may be either $v_i(1) = 1$ or $v_i(1) = \delta > 0$, where, $\delta \ll 1$ (e.g., $\delta = \frac{1}{n^2}$). We say that the type of i is 1 in the first case, and δ otherwise. Moreover, each agent has valuation $v_i(0) = 0$ if the project is not implemented. The designer would like to implement the project only if at least c individuals have type 1. In other words, the designer would like to implement the *public project function* f that returns 1 if $\sum_i v_i(1) \geq c$, and returns 0 otherwise. This has been introduced by [Jackson and Moulin, 1992] and it is a basic and very well studied problem in economics and computer science (see, e.g., [Apt and Estévez-Fernández, 2009]).

Every mechanism for the public project problem queries the agents about their type for the project. We will denote as $b_i \in \{\delta, 1\}$ the declaration (bid) of the agent i when queried. Indeed, since there are only two types, every query that is able to distinguish subsets of agent's types, directly reveals the type, and every query that is not able to make this distinction is equivalent to not querying at all. Hence, we can safely assume that either the mechanism does not query an agent, or, if it does, it just queries once to directly reveal the type. We will denote as $\pi\colon \mathbb{N}_{\geq 1} \to 2^N$ the order in which agents are queried, where $\pi(t)$ denotes the subset of agents queried at time t. As stated above, for every agent i there is at most one time step t_i for which $i \in \pi(t_i)$. Moreover, given a query order π and an agent i queried at time t_i according to this order, we will denote with n_i and k_i the number of agents that have been queried, and the ones that declared to have type 1, before i is queried for her type, respectively. That is, $n_i = \left|\bigcup_{t<t_i} \pi(t)\right|$ and $k_i = \left|\left\{j \in \bigcup_{t<t_i} \pi(t) \mid b_j = 1\right\}\right|$.

Lemma 4. *For every $\varepsilon \in [0, 1)$, every ε-OSP mechanism that implements the public project function has to verify every queried customer i as long as $n_i \leq n + k_i - c - 1$ and $k_i \leq c - 2$. Moreover, if $b_i = 1$, then i must be verified even if $n_i = n + k_i - c$ or $k_i = c - 1$.*

Proof. Let us first suppose that i has type 1. Observe that if $k_i > c - 2$, then it is obviously dominant for i to declare her type truthfully. Indeed, since the mechanism implements the public project function, it returns 1 in this case, and i will achieve her maximum possible utility, regardless of others' declarations. Similarly, if $n - n_i - 1 < c - k_i$, then it is obviously dominant for i to declare her type truthfully, otherwise she will achieve the minimum possible utility, i.e., 0, regardless of what the remaining agents declare. However, if $c + n_i + 1 - n \leq k_i \leq c - 2$, then truthtelling ceases to be obviously dominant: the worst outcome for i when she is truthful is achieved when no remaining individual declares type 1, so that the mechanism outcomes 0 (since there are at most $c-1$ customers with type 1), and this outcome has value 0 for i; instead, by declaring δ, the best outcome would occur when at least $c - k_i$ among the remaining agents (that are $n - n_i - 1 \geq c - k_i$) declare type 1, so that the service is activated, by giving to i utility $1 > 0 + \varepsilon$.

Hence, in order for the mechanism to be ε-OSP it is necessary to verify the agent i that declares type δ whenever $k_i \in [c + n_i + 1 - n, c - 2]$. A similar argument also can be adopted when i's type is δ, from which we achieve that, if the mechanism is ε-OSP, then the agent i that declares 1 must be verified whenever $k_i \in [c + n_i - n, c - 1]$. □

The order π may be defined by the mechanism or by nature (that is, the mechanism processes agents as they come). Unless differently specified, we do not make below any difference about this feature and our results hold no matter the source of π. The *history* at time $t \geq 1$ is $H_t = ((\pi(1), \mathbf{b}_{\pi(1)}), \ldots, (\pi(t), \mathbf{b}_{\pi(t)}))$, where for a subset of agents $S \subseteq N$, $\mathbf{b}_S = (b_i)_{i \in S}$. We will denote with $\mathcal{H}_t$ the set of all possible histories at time t and with $N_\pi^{t-1} = \bigcup_{j<t} \pi(j)$ the set of agents queried by π before time t. Moreover we set $\mathcal{H}_0 = H_0 = \emptyset$. A *selection rule* $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_l)$, $l \leq n$, where each $\sigma_t\colon \mathcal{H}_{t-1} \to \Delta(2^N)$, associates to each possible history H_{t-1} a probability distribution $\sigma_t(H_{t-1})$ over the subsets of agents in $N \setminus N_\pi^{t-1}$. Roughly speaking, the selection rule specifies how the mechanism (nature, resp.) selects which players will be queried next. This definition allows us to represent every selection rule, even adaptive ones (in which players are selected based also on the bids submitted by previously queried agents).

The *uniform selection* rule U returns, for every t and for every history, the uniform distribution over non-queried agents. Given c and a selection rule $\boldsymbol{\sigma}$, for a type profile $\mathbf{t}$ such that $|\{i\colon t_i = 1\}| = c$, we let $\tau_{\boldsymbol{\sigma},c}(\mathbf{t})$ be the random variable that measures the number of agents that have been verified by the mechanism on type profile $\mathbf{t}$. When clear from the context, we omit c from the subscript. Next we show that for every selection rule $\boldsymbol{\sigma}$ there is an instance $\mathbf{t}$ in which $\boldsymbol{\sigma}$ performs worse than the uniform selection rule, in terms of the expected number of agents verified $E[\tau_{\boldsymbol{\sigma}}(\mathbf{t})]$.

If a selection rule queries more than one agent at the same time, then it can only reach the thresholds of Lemma 4 later than when these queries are serialized. Moreover, as highlighted in [Bade and Gonczarowski, 2017; Mackenzie, 2017], serialization does not affect the OSP-ness of the mechanism.

Lemma 5. *For $\varepsilon \in [0, 1)$, $c > 0$ and selection rule $\boldsymbol{\sigma}$ for a ε-OSP mechanism that implements the public project function, there is $\mathbf{t}$ s.t. $|\{i\colon t_i = 1\}| = c$ and $E[\tau_{\boldsymbol{\sigma}}(\mathbf{t})] \geq E[\tau_U(\mathbf{t})]$.*

Proof. As stated above, we can assume that at each time step $\boldsymbol{\sigma}$ assigns positive probability only to singletons.

Let P be the uniform distribution on the type profiles $\mathbf{t}$ such that $|\{i\colon t_i = 1\}| = c$, i.e. $P(\mathbf{t}) = \left(\binom{n}{c}(n-c)!c!\right)$ if $|\{i\colon t_i = 1\}| = c$ and $P(\mathbf{t}) = 0$ o.w.. For every (t, H_{t-1}) and for agent i selected at time t, $\Pr_P(t_i = 1 \mid H_{t-1}) = \frac{c-k_i}{n-n_i}$.

We prove by induction on i that $E_{\mathbf{t}\sim P}[\tau_{\boldsymbol{\sigma}}(\mathbf{t}_{N\setminus N_\pi^{n-i}}) \mid H_{n-i}] \geq E_{\mathbf{t}\sim P}[\tau_U(\mathbf{t}_{N\setminus N_\pi^{n-i}}) \mid H_{n-i}]$ for every H_{n-i} (and thus $\mathbf{t}_{N_\pi^{n-i}}$). Since $H_0 = \emptyset$, we have that $E_{\mathbf{t}\sim P}[\tau_{\boldsymbol{\sigma}}(\mathbf{t})] \geq E_{\mathbf{t}\sim P}[\tau_U(\mathbf{t})]$. The lemma follows because it must exist $\mathbf{t}^\star$ s.t. $|\{i\colon t_i^\star = 1\}| = c$ and $E[\tau_{\boldsymbol{\sigma}}(\mathbf{t}^\star)] \geq E[\tau_U(\mathbf{t}^\star)]$.

The base case is $i = 1$. Observe that every selection rule will query exactly the same agent, i.e., the one that has not been queried in H_{n-1}. For the given history, let k denote the number of agents who declared 1. It must be that either $k = c$ or $k = c-1$ (since we are only considering profiles with exactly c agents with type 1). According to Lemma 4, in the first case the last agent must not be verified, whereas in the last case it is necessary to verify the last agent only if she declares 1. However, since this choice is independent from the selection rule and from $\mathbf{t}_{N_\pi^{n-1}}$, the claim trivially holds.

Assume now that the claim holds for $i-1$. We prove it also for i. Consider a history H_{n-i}. If at least c agents declaring type 1 have been queried, or the number of those whose type is still unknown is too low for reaching the threshold c, then no further verification needs to be made regardless of the selection rule adopted in the remaining steps. Hence, the claim trivially holds. If instead $k \leq c-1$ customers declared 1 and there are exactly $c-k$ non-queried agents, then for every $\mathbf{t}$ such that $|\{i\colon t_i = 1\}| = c$, it will occur that all non-queried agents have type 1 and thus, according to Lemma 4 they must be all verified, and the claim trivially holds. Consider now the case that $k \leq c-1$ customers declared 1 and there are $p \geq c-k+1$ agents that have not been queried yet. If $\sigma_{n-i+1}(H_{n-i}) = U(H_{n-i})$, then the claim directly follows from the inductive hypothesis. Otherwise, let $S = N \setminus N_\pi^{n-i}$. By Lemma 4 the agent selected at the $(n-i+1)$-th time step will be surely verified. Then, by ε-OSPness and the inductive hypothesis, $E_{\mathbf{t}\sim P}[\tau_{\boldsymbol{\sigma}}(\mathbf{t}_{N\setminus N_\pi^{n-i}}) \mid H_{n-i}] \geq 1 + \sum_{z\in S}\sigma_{n-i+1}(H_{n-i})(z)\cdot\sum_{\beta\in\{\delta,1\}}\Pr_P(t_z = \beta)E_{\mathbf{t}\sim P}[\tau_U(\mathbf{t}_{N\setminus N_\pi^{n-i}}) \mid H_{n-i}\cup(z,\beta)]$.

By anonymity of U, $E[\tau_U(\mathbf{t}) \mid H_t]$ depends only on how many agents have type 1 in H_t. Then, $E_{\mathbf{t}\sim P}[\tau_{\boldsymbol{\sigma}}(\mathbf{t}_{N\setminus N_\pi^{n-i}}) \mid H_{n-i}] = E_{\mathbf{t}\sim P}[\tau_U(\mathbf{t}_{N\setminus N_\pi^{n-i}}) \mid H_{n-i}]$. □

We can now state the main theorem of this section.

Theorem 6. *For all ε-OSP mechanisms implementing the public project function, with $\varepsilon \in [0,1)$, there is an instance on which the mechanism verifies in expectation $n - o(n)$ agents.*

Proof. Let $c = 1 + \sqrt{n-1}$. We next show that if the mechanism adopts the uniform selection rule, then for every $\mathbf{t}$ such that $|\{i\colon t_i = 1\}| = c$, $E[\tau_U(\mathbf{t})] \geq n - o(n)$. The theorem then follows by merging this result with Lemma 5; in fact,

$$\begin{aligned} E[\tau_U(\mathbf{t})] &\geq (n-c-1)\cdot(1 - \Pr(\tau_U(\mathbf{t}) < n-c-1)) \\ &= (n-o(n))\,(1 - \Pr(\tau_U(\mathbf{t}) < n-c-1))\,. \end{aligned}$$

We next prove that $\Pr(\tau_U(\mathbf{t}) < C) = o(1)$. For $\tau_U(\mathbf{t})$ to be less than $n-c-1$ it must be the case that $c-1$ customers with value 1 have been selected among the first $n-c-2$ queried agents. Indeed, since a mechanism that implements the public project function must query all agents until either c agents have declared 1 or the number of agents left to query is insufficient to reach this threshold, the $(n-c-1)$-th agent must be necessarily queried, and, by Lemma 4, verified. Observe that there are $\binom{n-c-2}{c-1}$ ways to query customers so that this property is satisfied among the $\binom{n}{c}$ ways in which the c customers with value 1 can be queried. Since we are using the uniform selection rule, these arrangements have the same probability, and thus $\Pr(\tau_U(\mathbf{t}) < n-c-1) = \binom{n-c-2}{c-1}/\binom{n}{c} = o(1)$. □

We now prove that Theorem 6 is essentially tight. For a social choice function f, the *sensitivity* of f is the smallest integer d such that $|f((t_i,\mathbf{t}_{-i}),s) - f((t_i',\mathbf{t}_{-i}),s)| \leq \frac{d}{n}$ for every i, every $t_i, t_i' \in T_i$, every $\mathbf{t}_{-i} \in T_{-i}$, and every s.

Fix $c = o(n)$ and $\varepsilon > 0$ such that $\frac{4dc}{\varepsilon} = o(n)$ and let $\beta = \frac{n\varepsilon}{2dc}$. Consider the following mechanism, that we will name $\mathcal{M}_\beta$: query agents for their type in sequential order; given a declaration profile $\mathbf{b} \in D$, choose the outcome $s \in \mathcal{S}$ according to the probability distribution $M^\beta(\mathbf{b}, s) = \frac{e^{\beta f(\mathbf{b},s)}}{\sum_{z\in S} e^{\beta f(\mathbf{b},z)}}$; for the first $n-c$ agents, verify if her declared type b_i coincides with her real type t_i. We have the following theorem, whose proof mimics [Nissim *et al.*, 2012, Thm. 2].

Theorem 7. *For every social choice function f, if $|S| \leq e^{o(n)}$, then the $\mathcal{M}_\beta$ mechanism is 2ε-OSP in expectation*[2] *and it implements f asymptotically.*

5 Conclusions

[Li, 2017] formally proved that OSP is the "right" definition of truthfulness for a kind of "bounded rational" agents, where the kind of bounded rationality (i.e., those who have limited who lack contingent reasoning skills). is exactly the one observed in many experimental settings. This motivated the analysis of what can and cannot be done with these partially rational agents. This investigation led both positive [Li, 2017; Pycia and Troyan, 2016; Ferraioli *et al.*, 2018] and negative results [Ashlagi and Gonczarowski, 2015; Bade and Gonczarowski, 2017; Ferraioli and Ventre, 2017] The latter encourage to find alternative methods to achieve the OSP property (just as impossibility theorem by Gibbard and Satterwhaite [Gibbard, 1973; Satterthwaite, 1975] encouraged research about implementation with payments). From this point of view, our results indeed can be seen as providing an useful engineering tool to achieve obvious strategyproofness whenever one can afford costly verification or set large penalties, and an impossibility result otherwise.

However, at light of the many negative results, it would also be interesting to investigate mechanism design for other (possibly, less stringent) notions of bounded rationality.

In Section 4, we focused on the binary public project problem. The simplicity of this problem has two advantages: (i)

[2]For the formal definition of ε-OSP in expectation mechanisms, we refer to the full version of the paper [Ferraioli and Ventre, 2018].

it makes our result stronger, since we are giving a negative result; (ii) it improves the readability of the proof. However, our proof uses the structure of the problem only to prove that: (i) one needs to verify almost every queried agent until you find a solution; (ii) there is an instance for which it is unlikely to find a solution after few queries. But these properties are enjoyed not only by the public project problem, but also by its combinatorial counterpart and many other different problems (e.g., facility location). It would, however, be interesting to find settings in which an OSP mechanism with partial probabilistic verification exists that verifies only few agents.

References

[Adamczyk *et al.*, 2015] Marek Adamczyk, Allan Borodin, Diodato Ferraioli, Bart de Keijzer, and Stefano Leonardi. Sequential posted price mechanisms with correlated valuations. In *WINE 2015*, pages 1–15, 2015.

[Apt and Estévez-Fernández, 2009] Krzysztof R. Apt and Arantza Estévez-Fernández. Sequential pivotal mechanisms for public project problems. In *SAGT 2009*, pages 85–96, 2009.

[Ashlagi and Gonczarowski, 2015] Itai Ashlagi and Yannai A Gonczarowski. No stable matching mechanism is obviously strategy-proof. *arXiv preprint arXiv:1511.00452*, 2015.

[Babaioff *et al.*, 2014] Moshe Babaioff, Nicole Immorlica, Brendan Lucier, and S Matthew Weinberg. A simple and approximately optimal mechanism for an additive buyer. In *FOCS 2014*, pages 21–30, 2014.

[Bade and Gonczarowski, 2017] Sophie Bade and Yannai A. Gonczarowski. Gibbard-satterthwaite success stories and obvious strategyproofness. In *EC 2017*, page 565, 2017.

[Brânzei and Procaccia, 2015] Simina Brânzei and Ariel D Procaccia. Verifiably truthful mechanisms. In *ICTCS 2015*, pages 297–306, 2015.

[Caragiannis *et al.*, 2012] Ioannis Caragiannis, Edith Elkind, Mario Szegedy, and Lan Yu. Mechanism design: from partial to probabilistic verification. In *EC 2012*, pages 266–283, 2012.

[Chawla *et al.*, 2010] Shuchi Chawla, Jason D Hartline, David L Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *STOC 2010*, pages 311–320, 2010.

[Ferraioli and Ventre, 2017] Diodato Ferraioli and Carmine Ventre. Obvious strategyproofness needs monitoring for good approximations. In *AAAI 2017*, pages 516–522, 2017.

[Ferraioli and Ventre, 2018] D. Ferraioli and C. Ventre. Probabilistic Verification for Obviously Strategyproof Mechanisms. *ArXiv e-prints*, 2018.

[Ferraioli *et al.*, 2015] Diodato Ferraioli, Carmine Ventre, and Gabor Aranyi. A mechanism design approach to measure awareness. In *AAAI 2015*, pages 886–892, 2015.

[Ferraioli *et al.*, 2016] Diodato Ferraioli, Paolo Serafino, and Carmine Ventre. What to verify for optimal truthful mechanisms without money. In *AAMAS 2016*, pages 68–76, 2016.

[Ferraioli *et al.*, 2018] Diodato Ferraioli, Adrian Meier, Paolo Penna, and Carmine Ventre. On the approximation guarantee of obviously strategyproof mechanisms. 2018.

[Fotakis *et al.*, 2016] Dimitris Fotakis, Christos Tzamos, and Manolis Zampetakis. Mechanism design with selective verification. In *EC 2016*, pages 771–788, 2016.

[Fotakis *et al.*, 2017] Dimitris Fotakis, Piotr Krysta, and Carmine Ventre. Combinatorial auctions without money. *Algorithmica*, 77(3):756–785, 2017.

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

[Hartline and Roughgarden, 2009] Jason D Hartline and Tim Roughgarden. Simple versus optimal mechanisms. In *EC 2009*, pages 225–234, 2009.

[Jackson and Moulin, 1992] Matthew Jackson and Herve Moulin. Implementing a public project and distributing its cost. *Journal of Economic Theory*, 57(1):125–140, 1992.

[Li, 2017] Shengwu Li. Obviously strategy-proof mechanisms. *American Economic Review*, 107(11):3257–87, 2017.

[Mackenzie, 2017] Andrew Mackenzie. A revelation principle for obviously strategy-proof implementation. 2017.

[Nissim *et al.*, 2012] Kobbi Nissim, Rann Smorodinsky, and Moshe Tennenholtz. Approximately optimal mechanism design via differential privacy. In *ITCS 2012*, pages 203–213, 2012.

[Penna and Ventre, 2014] Paolo Penna and Carmine Ventre. Optimal collusion-resistant mechanisms with verification. *Games and Economic Behavior*, 86:491–509, 2014.

[Procaccia and Tennenholtz, 2013] Ariel D. Procaccia and Moshe Tennenholtz. Approximate mechanism design without money. *ACM Trans. Economics and Comput.*, 1(4):18:1–18:26, 2013.

[Pycia and Troyan, 2016] Marek Pycia and Peter Troyan. Obvious dominance and random priority. 2016.

[Sandholm and Gilpin, 2003] Tuomas Sandholm and Andrew Gilpin. Sequences of take-it-or-leave-it offers: Near-optimal auctions without full valuation revelation. In *AMEC 2003*, pages 73–91, 2003.

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

[Zhang and Levin, 2017] Luyao Zhang and Dan Levin. Partition obvious preference and mechanism design: Theory and experiment. 2017.