# Obvious strategyproofness needs monitoring for good approximations

**Diodato Ferraioli**
DIEM, Università degli Studi di Salerno, Italy
dferraioli@unisa.it

**Carmine Ventre**
CSEE, University of Essex, UK
carmine.ventre@gmail.com

## Abstract

Obvious strategyproofness (OSP) is an appealing concept as it allows to maintain incentive compatibility even in the presence of agents that are not fully rational, e.g., those who struggle with contingent reasoning (Li 2015). However, it has been shown to impose some limitations, e.g., no OSP mechanism can return a stable matching (Ashlagi and Gonczarowski 2015).

We here deepen the study of the limitations of OSP mechanisms by looking at their approximation guarantees for basic optimization problems paradigmatic of the area, i.e., machine scheduling and facility location. We prove a number of bounds on the approximation guarantee of OSP mechanisms, which show that OSP can come at a significant cost. However, rather surprisingly, we prove that OSP mechanisms can return optimal solutions when they use monitoring — a novel mechanism design paradigm that introduces a mild level of scrutiny on agents' declarations (Kovács, Meyer, and Ventre 2015).

## Introduction

Algorithmic Mechanism Design (AMD) is by now an established research area in computer science that aims at conceiving algorithms resistant to selfish manipulations. As the number of parties (a.k.a., agents) involved in the computation increases, there is, in fact, the need to realign their individual interests with the designer's. Truthfulness is the chief concept to achieve that: in a truthful mechanism, no selfish and *rational* agent has an interest to misguide the mechanism. A valid question of recent interest is, however, how easy it is for the selfish agents to understand that it is useless (and possibly costly) to strategize against the truthful mechanism at hand.

Recent research has come up with different approaches to deal with this question. Some authors (Sandholm and Gilpin 2003; Babaioff et al. 2014; Chawla et al. 2010; Adamczyk et al. 2015) suggest to focus on "simple" mechanisms; e.g., in posted-price mechanisms one's own bid is immaterial for the price paid to get some goods of interest – this should immediately suggest that trying to play the mechanism is worthless no matter the cognitive abilities of the agents. However, in such a body of work, this property remains unsatisfactorily vague. An orthogonal approach is that of verifiably truthful mechanisms (Brânzei and Procaccia 2015), wherein agents can run some algorithm to effectively check that the mechanism is incentive compatible. Nevertheless, these verification algorithms can run for long (i.e., time exponential in the input size) and are so far known only for quite limited scenarios. Importantly, moreover, they seem to transfer the question from the mechanism itself to the verification algorithm.

Li (2015) has recently formalized the aforementioned idea of simple mechanisms, by introducing the concept of Obviously Strategy-Proof (OSP) mechanisms. This notion stems from the observation that the very same mechanism can be more or less truthful in practice depending on the implementation details. For example, in lab experiments, Vickrey's famous second-price mechanism results to be "less" truthful when implemented via a sealed-bid auction, and "more" truthful when run via an ascending auction. The quite technical definition of OSP formally captures how implementation details matter by looking at a mechanism as an extensive-form game; roughly speaking, OSP demands that strategyproofness holds among subtrees of the game (see below for a formal definition). An important validation for the 'obviousness' is further provided by Li (2015) via a characterization of these mechanisms in terms of agents with limited cognitive abilities (i.e., agents with limited skills in contingent reasoning). Specifically, Li shows that a strategy is obviously dominant if and only if these "limited" agents can recognize it as such.

Nevertheless, for all its significant aspects, there appear to be hints that the notion of OSP mechanisms might be too restrictive. Ashlagi and Gonczarowski (2015) prove, for example, that no OSP mechanism can return a stable matching – thus implying that the Gale-Shapley matching algorithm is not OSP despite its apparent simplicity.

**Our contribution.** We investigate the power of OSP mechanisms in more detail from a theoretical computer science perspective. In particular, we want to understand the quality of approximate solutions that can be output by OSP mechanisms. To answer this question, we focus on two fundamental optimization problems, machine scheduling (Archer and Tardos 2001) and facility location (Moulin 1980), arguably (among) the paradigmatic problems in AMD.

For the former problem, we want to compute a schedule of jobs on selfish related machines (i.e., machines with job-independent speeds) so to minimize the makespan. For this single-dimensional problem, it is known that a truthful PTAS is possible (Christodoulou and Kovács 2013). In contrast, we show that there is no better than 2-approximate OSP mechanism for this problem independently from the running time of the mechanism.

For the facility location problem, we want to determine the location of a facility on the real line given the preferred locations of $n$ agents. The objective is to minimize the social cost, defined as the sum of the individual agents' distances between their preferred location and the facility's. Moulin (1980) proves that the optimal mechanism, that places the facility on the median of the reported locations, is truthful without money (i.e., the mechanism does not pay or charge the agents). OSP mechanisms without money turn out to be much weaker than that. We prove in fact a tight bound of $n-1$. Interestingly, a linear bound also holds for mechanisms that use money, thus showing that transfers are not that useful to enforce OSP.

However, a surprising connection of OSP mechanisms with a novel mechanism design paradigm – called *monitoring* – allows us to prove strong positive results. Building upon the notion of mechanisms with verification (Nisan and Ronen 2001; Penna and Ventre 2014), Kovács, Meyer, and Ventre (2015) introduce the idea that a mechanism can check the declarations of the agents at running time and guarantee that those who overreported their costs end up paying the exaggerated costs. This can be enforced whenever costs can be easily measured and certified. For example, a mechanism can force a machine that in her declaration has augmented her running time to work that long by keeping her idle for the difference between real and reported running time. We prove that, for both our problems of interest, there is an optimal OSP mechanism with monitoring. Both our mechanisms have quite interesting and distinctive features. The construction of the mechanism for machine scheduling can use any algorithm for the problem as a black box, thus implying that there is PTAS that is OSP. The mechanism for facility location, instead, is the first direct-revelation mechanism that is OSP – previously known constructions relied on indirect mechanisms querying agents to find out more about their types. Our constructions are based upon the first-price (truthful) mechanism (with monitoring) recently designed in (Serafino, Vidali, and Ventre 2016). Our results effectively show how it is possible to modify this mechanism to allow OSP implementations.

## Preliminaries

**Mechanisms and Strategy-proofness.** In this work we consider a classical mechanism design setting, in which we have a set of outcomes $\mathcal{O}$ and $n$ selfish agents. Each agent $i$ has a *type* $t_i \in D_i$, where $D_i$ is defined as the *domain* of $i$. The type $t_i$ is *private knowledge* of agent $i$. Moreover, each selfish agent $i$ has a *cost function* $c_i \colon D_i \times \mathcal{O} \to \mathbb{R}$. For $t_i \in D_i$ and $X \in \mathcal{O}$, $c_i(t_i, X)$ is the cost paid by agent $i$ to implement $X$ when her type is $t_i$.

A *mechanism* consists of a protocol whose goal is to determine an outcome $X \in \mathcal{O}$. To this aim, the mechanism is allowed to interact with agents. During this interaction, agent $i$ is observed to take *actions* (e.g., saying yes/no); these actions may depend on her presumed type $b_i \in D_i$ that can be different from the real type $t_i$ (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. For a mechanism $\mathcal{M}$, we let $\mathcal{M}(\mathbf{b})$ denote the outcome returned by the mechanism when agents take actions according to their presumed types $\mathbf{b} = (b_1, \ldots, b_n)$. Usually, this outcome is given by a pair $(f, \mathbf{p})$, where $f = f(\mathbf{b})$ (termed *social choice function* or, simply, algorithm) maps the actions taken by the agents according to $\mathbf{b}$ to a feasible solution for the problem at the hand (e.g., an allocation of jobs to machines that enjoys particular properties), and $\mathbf{p} = \mathbf{p}(\mathbf{b}) = (p_1(\mathbf{b}), \ldots, p_n(\mathbf{b})) \in \mathbb{R}^n$ maps the actions taken by the agents according to $\mathbf{b}$ to *payments* from the mechanism to each agent $i$. Note that the $p_i$'s can be positive (meaning that the mechanism will pay the agents) or negative (meaning that the agents will pay the mechanism).

A mechanisms is said *without money* if $p_i(\mathbf{b}) = 0$ for every agent $i$ and every profile $\mathbf{b} \in D = D_1 \times \cdots \times D_n$. Our definitions below do naturally extend to this case by considering null payments.

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 $b_i \in D_i$, it holds that $c_i(t_i, \mathcal{M}(t_i, \mathbf{b}_{-i})) \leq c_i(t_i, \mathcal{M}(b_i, \mathbf{b}_{-i}))$, where $t_i$ is the true type of $i$. That is, in a strategy-proof mechanism the actions taken according to the true type are dominant for each agent.

Moreover, a mechanism $\mathcal{M}$ is said to satisfy *voluntary participation* if for every $i$ and every $\mathbf{b}_{-i}$, it holds that $c_i(t_i, \mathcal{M}(t_i, \mathbf{b}_{-i})) \leq 0$.

**Obvious Strategy-proofness.** Let us now formally define the concept of obviously strategy-proof mechanism. This concept has been introduced in (Li 2015). The original definition turns out to be very general and, consequently, quite complex. For this reason, in this work we follow Ashlagi and Gonczarowski (2015) and rephrase this definition for our setting of interest. Note that we focus on deterministic mechanisms only.

We being by formally modeling how a mechanism works and subsequently give some intuition behind the mathematical definition. Specifically, we have that an *extensive-form mechanism* $\mathcal{M}$ is defined by a directed tree $(V, E)$ such that:

- every leaf $\ell$ of the tree is labeled by a possible outcome $X(\ell) \in \mathcal{O}$ of the mechanism;
- every internal vertex $u \in V$ is labeled by a subset $S(u) \subseteq [n]$ of agents;
- every edge $e = (u, v) \in E$ is labeled by a subset $T(e) \subseteq D$ of type profiles such that:
  - the subsets 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 subsets of profiles that label the edges outgoing from a non-root vertex $u$ is equal to the subset of profiles that label 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 $T$;
- the union of the subsets 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 $(b_i)_{i\in S(u)} = (b'_i)_{i\in S(u)}$, if $\mathbf{b}$ belongs to $T(u,v)$, then also $\mathbf{b}'$ must belong to $T(u,v)$.

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, the mechanism concurrently interacts with agents in $S(u)$. Different edges outgoing from 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 agents in $S(u)$ 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 $T(e)$. 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 agents in $S(u)$ can only inform about types of agents in $S(u)$ and not about the type of the remaining agents (that are completely unknown to agents in $S(u)$). The execution ends when we reach a leaf $\ell$ of the tree. In this case, the mechanism returns the outcome that labels $\ell$.

Observe that, according to the definition above, 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 $\mathbf{b} \in T(\phi(u),u)$. 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 $\mathbf{b}' \in T(u,v')$.

We are now ready to define obvious strategy-proofness. An extensive-form mechanism $\mathcal{M}$ is *obviously strategy-proof (OSP)* if for every agent $i$, for every vertex $u$ such that $i \in S(u)$, for every $\mathbf{b}_{-i}, \mathbf{b}'_{-i}$, and for every $b_i \in D_i$ such that $(t_i, \mathbf{b}_{-i})$ and $(b_i, \mathbf{b}'_{-i})$ are compatible with $u$, but diverge at $u$, it holds that $c_i(t_i, \mathcal{M}(t_i, \mathbf{b}_{-i})) \leq c_i(t_i, \mathcal{M}(b_i, \mathbf{b}'_{-i}))$. Roughly speaking, an obvious strategy-proof mechanism requires that, at each time step agent $i$ is asked to take a decision that depends on her type, the worst cost that she can pay if at this time step she behaves according to her true type is at least the same as the best cost achievable by behaving as she had a different type.

Hence, if a mechanism is obviously strategy-proof, then it is also strategy-proof. Indeed, the latter requires that truthful behavior is a dominant strategy when agents know the entire type profile, whereas the former requires that it continues to be a dominant strategy even if agents have only a partial knowledge of profiles[1], limited to what they observed in the mechanism up to the time they are called to take their choices.

We say that an extensive-form mechanism is *trivial* if for every vertex $u \in V$ and for every two type profiles $\mathbf{b}, \mathbf{b}'$, it holds that $\mathbf{b}$ and $\mathbf{b}'$ do *not* diverge at $u$. That is, a mechanism is trivial if it never requires that agents take actions that depend on their type. Observe that if a mechanism $\mathcal{M}$ is not trivial, then every path from the root to one leaf goes through a vertex $u^\star$ such that there are two type profiles $\mathbf{b}, \mathbf{b}'$ that diverge at $u^\star$. Since $\mathbf{b} \neq \mathbf{b}'$, then there exists at least one agent $i^\star$ such that $b_{i^\star} \neq b'_{i^\star}$. Moreover, by our definition of extensive-form mechanism, it must be the case that $i^\star \in S(u^\star)$. For this reason, we call $i^\star$ as the *divergent agent* for the mechanism $\mathcal{M}$. Note that the divergent agent takes a decision that depends on her own type before any other agents revealed any information about their own type. For this reason, in order to prove that a mechanism is not obviously strategy-proof, it is sufficient to show that there are two type profiles $\mathbf{b}, \mathbf{b}'$ with $b_{i^\star} \neq b'_{i^\star}$ such that they diverge at $u^\star$, and $c_{i^\star}(b_{i^\star}, \mathcal{M}(\mathbf{b})) > c_{i^\star}(b_{i^\star}, \mathcal{M}(\mathbf{b}'))$.

Let us state two further properties of obvious strategy-proofness, that turn out to be very useful in the rest of the paper. First, it is not hard to see that if $\mathcal{M}$ is OSP when the type profile is taken from $D$, then it continues to enjoy this property even if the types are only allowed to be selected from $D' = D'_1 \times \cdots \times D'_n$, where $D'_i \subseteq D_i$. Moreover, let us define $\mathcal{M}'$ obtained from $\mathcal{M}$ by *pruning* the paths involving actions corresponding to types in $D \setminus D'$. If $\mathcal{M}$ is OSP, then also $\mathcal{M}'$ enjoys this property (Li 2015).

**Monitoring.** Let $\mathcal{M}(\mathbf{b})$ denote the outcome returned by mechanism $\mathcal{M} = (f, \mathbf{p})$ when agents take actions according to $\mathbf{b}$. Commonly, the cost paid by agent $i$ to implement $\mathcal{M}(\mathbf{b})$ is defined as a quasi-linear combination of agent's true cost[2] $t_i(f(\mathbf{b}))$ and payment $p_i(\mathbf{b})$, i.e., $c_i(t_i, \mathcal{M}(\mathbf{b})) = t_i(f(\mathbf{b})) - p_i(\mathbf{b})$. This approach disregards the agent's declaration for evaluating her cost.

In mechanisms with monitoring the usual quasi-linear definition is maintained but costs paid by the agents are more strictly tied to their declarations (Kovács, Meyer, and Ventre 2015). Specifically, in a mechanism with monitoring $\mathcal{M}$, the bid $b_i$ is a lower bound on agent $i$'s cost for $f(b_i, \mathbf{b}_{-i})$, so an agent is allowed to have a real cost higher than $b_i(f(\mathbf{b}))$ but not lower. Formally, we have $c_i(t_i, \mathcal{M}(\mathbf{b})) = \max\{t_i(f(\mathbf{b})), b_i(f(\mathbf{b}))\} - p_i(\mathbf{b})$.

We next describe two specific problems of interest.

[1] In fact, OSP implies – but is not equivalent to – weakly group strategy-proofness (Li 2015).

[2] Note that $t_i(f(\mathbf{b}))$ depends only on her type and the outcome of the social choice function.

**Machine scheduling.** Here, we are given a set of $m$ different jobs to execute and the $n$ agents control related machines. That is, agent $i$ has a job-independent processing time $t_i$ per unit of job (equivalently, an execution speed $1/t_i$ that is independent from the actual jobs). Therefore, the social choice function $f$ must choose a possible schedule $f(\mathbf{b}) = (f_1(\mathbf{b}), \ldots, f_n(\mathbf{b}))$ of jobs to the machines, where $f_i(\mathbf{b})$ denotes the job load assigned to machine $i$ when agents take actions according to $\mathbf{b}$. The cost that agent $i$ faces for the schedule $f(\mathbf{b})$ is $t_i(f(\mathbf{b})) = t_i \cdot f_i(\mathbf{b})$. Note that our mechanisms for machine scheduling will always pay the agents.

Monitoring can be readily implemented for this setting. In fact, monitoring means that those agents who have exaggerated their unitary processing time, i.e., they take actions according to $b_i > t_i$, can be made to process up to time $b_i$ instead of the true processing time $t_i$. For example, we could not allow any other operation in the time interval $[t_i, b_i]$ or charge $b_i - t_i$.

We focus on social choice functions $f^*$ optimizing the *makespan*, i.e.,

$$f^*(\mathbf{b}) \in \arg\min_{\mathbf{x}} \mathtt{mc}(\mathbf{x}, \mathbf{b}), \quad \mathtt{mc}(\mathbf{x}, \mathbf{b}) = \max_{i=1}^{n} b_i(\mathbf{x}).$$

We say that $f$ is $\alpha$-approximate if it returns a solution whose cost is a factor $\alpha$ away from the optimum.

**Facility location.** In the facility location problem, the type $t_i$ of each agent consists of her position on the real line. The social choice function $f$ must choose a position $f(\mathbf{b}) \in \mathbb{R}$ for the facility. The cost that agent $i$ pays for a chosen position $f(\mathbf{b})$ is $t_i(f(\mathbf{b})) = d(t_i, f(\mathbf{b})) = |t_i - f(\mathbf{b})|$. So, $t_i(f(\mathbf{b}))$ denotes the distance between $t_i$ and the location of the facility computed by $f$ when agents take actions according to $\mathbf{b}$.

We can implement monitoring also in this setting whenever evidences of the distance can be provided (and cannot be counterfeited). In fact, in this context, monitoring means that $t_i(f(\mathbf{b})) = \max\{d(t_i, f(\mathbf{b})), d(b_i, f(\mathbf{b}))\}$. Therefore, once the evidence is provided, the mechanism can check whether $t_i(f(\mathbf{b})) < b_i(f(\mathbf{b}))$ and charge the agent the difference for cheating.

We focus on social choice functions $f^*$ optimizing the *social cost*, i.e.,

$$f^*(\mathbf{b}) \in \arg\min_{x \in \mathbb{R}} \mathtt{cost}(x, \mathbf{b}), \quad \mathtt{cost}(x, \mathbf{b}) = \sum_{i=1}^{n} b_i(x).$$

As above, we say that $f$ is $\alpha$-approximate if it returns a solution whose cost is at most $\alpha$ away from the optimum.

## Machine Scheduling

Let us start by showing that there is no OSP mechanism that satisfies voluntary participation and returns an assignment of jobs to machines whose makespan is at most twice the makespan of the optimal assignment. Interestingly, this is the same lower bound that Nisan and Ronen (2001) proved for the approximation ratio of strategy-proof mechanisms for *unrelated* machines, i.e., when it is not possible to express the processing time of jobs on machines as a product of jobs' load and machine's unit processing time. We wonder if a more deep relationship exists between OSP mechanisms for scheduling on related machines and strategy-proof mechanisms for scheduling on unrelated machines, and if one can improve the lower bound for the former problem in order to match the best known lower bounds for the latter, i.e., $1 + \phi \approx 2,61$ for general mechanisms (Koutsoupias and Vidali 2007), and $n$ for anonymous mechanisms (Ashlagi, Dobzinski, and Lavi 2012).

**Theorem 1.** *For every $\varepsilon > 0$, there is no $(2 - \varepsilon)$-approximate mechanism for the machine scheduling problem that is OSP without monitoring and satisfies voluntary participation.*

*Proof.* Let us consider the simple setting in which there are exactly two machines, that we denote with 0 and 1, and two equivalent jobs of unit length. We will denote with $t_0$ and $t_1$ the type, i.e., the job processing time, of machine 0 and 1, respectively. Suppose there is a $k$-approximate, with $k < 2$, OSP mechanism $\mathcal{M}$ that satisfies voluntary participation.

Since the mechanism is $k$-approximate, then it must be the case that: if $t_0 < \frac{t_1}{2k}$, then $\mathcal{M}$ assigns both jobs to machine 0; if $t_0 > 2k \cdot t_1$, then $\mathcal{M}$ assigns both jobs to machine 1; if $\frac{k}{2} \cdot t_1 < t_0 < \frac{2}{k} \cdot t_1$, then $\mathcal{M}$ assigns one job to each machine.

Moreover, since mechanism $\mathcal{M} = (f, \mathbf{p})$ is OSP, then it must be also strategy-proof. Archer and Tardos (2001) proved that a mechanism for the machine scheduling problem is strategy-proof and satisfies voluntary participation if and only if (i) the allocation of jobs to machine $i \in \{0, 1\}$ returned by $f$ when the type of the other machine is $t_{1-i}$ is *monotone*, i.e., if $f_i(t_i, t_{1-i}) \leq f_i(t_i', t_{1-i})$ whenever $t_i > t_i'$; (ii) the payment that the machine $i$ receives is

$$p_i(t_i, t_{1-i}) = t_i f_i(t_i, t_{1-i}) + \int_{t_i}^{\infty} f_i(x, t_{1-i}) dx.$$

In our setting, the monotonicity requirement implies that, for every $t_{1-i}$, there are $t' \in \left[\frac{t_{1-i}}{2k}, \frac{k}{2} \cdot t_{1-i}\right]$ and $t'' \in \left[\frac{2}{k} \cdot t_{1-i}, 2k \cdot t_{1-i}\right]$, such that machine $i$ is assigned both jobs if $t_i < t'$, only one job if $t' \leq t_i \leq t''$, and no jobs if $t_i > t''$. Hence, $p_i(t_i, t_{1-i}) = t' + t''$ if $t_i < t'$, $p_i(t_i, t_{1-i}) = t''$ if $t' \leq t_i \leq t''$, and $p_i(t_i, t_{1-i}) = 0$ otherwise.

Let us now restrict the domain of the agents to $D' = \{a, b\}^2$, with $b > k^2 a$. Let $\mathcal{M}'$ be the mechanism obtained by pruning $\mathcal{M}$ according to this restriction. As stated above, $\mathcal{M}'$ must be an OSP mechanism. Moreover, the approximation ratio of $\mathcal{M}'$ cannot be worse than the approximation ratio of $\mathcal{M}$. Hence, $\mathcal{M}'$ cannot be trivial (indeed, a trivial mechanism would have approximation ratio worse than $k$).

Let $i$ be the divergent agent of $\mathcal{M}'$. Clearly, $a$ and $b$ are the types in which $i$ diverges. Suppose that $t_i = a$. If $i$ behaves according to $t_i$, then it may be the case that the other agent behaves according type $a$ too. As showed above, in this case machine $i$ receives one job and payment $t'' \leq 2ka$. Hence, $c_i(a, \mathcal{M}(a, a)) \geq a - 2ka$. Suppose instead that $i$ behaves as if her type was $b$. It may be the case that the other agent behaves according type $b$ too. Then, machine $i$ still receives one job and a payment $t'' \geq \frac{2}{k} \cdot b$. Hence,

$$c_i(a, \mathcal{M}(b,b)) \leq a - \frac{2}{k} \cdot b < a - 2ka = c_i(a, \mathcal{M}(a,a)),$$

where we used that $b > k^2 a$. In words, the best cost paid by $i$ if she does not behave according to her true type can be lower than the worst cost she can pay if she behaves according to her true type. Then, the mechanism $\mathcal{M}'$ is not OSP, contradicting our hypothesis. □

We next show that it is possible to achieve better OSP mechanisms, if one allows monitoring. Specifically, we show in Theorem 2 below, that there is a mechanism that is OSP with monitoring that may use every (approximation) algorithm as a *black box*.

This positive result holds under the assumption that agents' domains are finite. Such an assumption is needed because we are dealing with an indirect mechanism, that is, the mechanism queries the agents to find out what their type is. An easy way to have a well-defined mechanism, that runs in finite time, is to require agents' domains to be finite as well so that each query can be linked to each value in the domain. For more general domains, there would need to be some way for the designer to link each query to more than one type. Such an assumption is done in (Li 2015) for his OSP mechanisms. Namely, Li (2015) assumes that mechanisms *admit a finite partition* of the set $D$ of type profiles in subsets $\Delta_1, \ldots, \Delta_s$ such that to every profile $\mathbf{b} \in \Delta_i$ there corresponds the same outcome. We remark that plugging in this assumption in Theorem 2 would be equivalent to requiring finite domains. Assume to the contrary that agents domains are infinite and let $f$ be the algorithm that minimizes the social cost for machine scheduling. Consider the case in which we have three players and two jobs of unitary weight. Take two bid vectors $\mathbf{b} = (a, c, d)$ and $\mathbf{b}' = (b, c, d)$ where $a < b < c$ and $d \in (a, b)$. Since the domains are infinite, we can always find $a, b, c$ and $d$ as from above. But then since $f$ minimizes the social welfare we have that $f(\mathbf{b}) \neq f(\mathbf{b}')$ as player 1 gets a job when declaring $a$ but does not when saying $b$. By repeating the same reasoning for all pairs $a, b \in D_1$ we reach the contradiction that the mechanism, defined upon $f$, does not admit a finite partition.

**Theorem 2.** *For every $\alpha$-approximate algorithm $f$ for the machine scheduling problem on related machines there is an $\alpha$-approximate mechanism for the same problem that is OSP with monitoring and satisfies voluntary participation.*

*Proof.* As suggested above, in order to implement an indirect OSP mechanism we assume that domains of agents are finite. To simplify exposition[3], we assume here that agents' types are discrete in the interval $[a, b]$ with discretization parameter $\delta > 0$, i.e., the domain of agent $i$ is $D_i = \{a, a + \delta, \ldots, b\}$.

Consider then the following mechanism $\mathcal{M}$:

- Let $p = a$ and let $A$ contain all the machines;
- While $A$ is not empty, do:
  - Concurrently ask each machine $i \in A$, if she accepts to execute any jobs for a payment of $p$ per unit of load;
  - If machine $i \in A$ does not accept, then we set $A = A \setminus \{i\}$, $b_i = p + \delta$;
  - Set $p = p - \delta$;
- Return the allocation $f(\mathbf{b}) = (f_1(\mathbf{b}), \ldots, f_n(\mathbf{b}))$;
- Assign to each machine a payment $p_i(\mathbf{b}) = f_i(\mathbf{b}) \cdot b_i$.

We will next prove that this mechanism is OSP, and thus, at the end of the algorithm, each $b_i$ corresponds to the real type of agent $i$. The claim then follows since $\mathcal{M}$ returns the allocation computed by an $\alpha$-approximate algorithm on the declared types.

In order to prove that $\mathcal{M}$ is OSP, consider agent $i$ and let $t_i$ be her processing time for unit of load. We say that agent $i$ adopts the *truthful strategy* if she accepts the offer as long as $p \geq t_i$, and refuses otherwise. Note that, if $i$ adopts the truthful strategy, then, at the end of the algorithm, $b_i = t_i$. We next show that for agent $i$ it is always convenient to adopt the truthful strategy, regardless of the decision taken in previous iterations by other machines. To this aim, let us recall that in a mechanism with monitoring the cost that $i$ pays, given the submitted type profile is $\mathbf{b}$, is

$$c_i(t_i, \mathcal{M}(\mathbf{b})) = \max\{t_i, b_i\} \cdot f_i(\mathbf{b}) - p_i(\mathbf{b}).$$

Suppose that $i$ adopts the truthful strategy, then for every $\mathbf{b}_{-i}$, it turns out that

$$c_i(t_i, \mathcal{M}(t_i, \mathbf{b}_{-i})) = t_i \cdot f_i(t_i, \mathbf{b}_{-i}) - p_i(t_i, \mathbf{b}_{-i}) = 0.$$

Suppose, instead, that $i$ adopts a different strategy. Since the mechanism stops to interact with machine $i$ as soon as she refuses an offer, we only need to care about the time in which this refusal occurs.

If the first refusal occurs when $i$ has been offered a payment $p \geq t_i$, then $b_i > t_i$ at the end of the algorithm. Hence, for every $\mathbf{b}_{-i}$,

$$c_i(t_i, \mathcal{M}(b_i, \mathbf{b}_{-i})) = b_i \cdot f_i(b_i, \mathbf{b}_{-i}) - p_i(b_i, \mathbf{b}_{-i}) = 0.$$

If the first refusal occurs when $i$ has been offered a payment $p < t_i - \delta$, then $t_i > b_i$ at the end of the algorithm. Hence, for every $\mathbf{b}_{-i}$,

$$c_i(t_i, \mathcal{M}(b_i, \mathbf{b}_{-i})) = t_i \cdot f_i(b_i, \mathbf{b}_{-i}) - p_i(b_i, \mathbf{b}_{-i}) > 0.$$

Thus, in both cases the best cost that $i$ can obtain by adopting a strategy different from the truthful one is not smaller than the worst cost that $i$ can obtain by adopting the truthful strategy, as desired. □

Since there is a PTAS for the allocation of jobs to related machines (Hochbaum and Shmoys 1988), then we have the following corollary. (To turn the PTAS algorithm into a PTAS mechanism that is OSP, we need to additionally assume that the domains have size polynomial in the input of the problem.)

**Corollary 1.** *There is an OSP mechanism with monitoring that computes the optimal scheduling of jobs to related machines. Moreover, there is an OSP mechanism with monitoring that is a PTAS for the same problem. Both mechanisms satisfy voluntary participation.*

[3] For more general finite domains, the mechanism would only require to take the structure of different $D_i$'s (e.g., the different "gaps" in the domain) into consideration.

## Facility Location without Money

**Theorem 3.** *For every $\varepsilon > 0$, there is no $(n-1-\varepsilon)$-approximate mechanism without money for the facility location problem that is OSP, even with monitoring.*

In order to prove Theorem 3, we first need to state the following lemma.

**Lemma 1.** *Consider a type profile $\mathbf{b}$ such that $b_i = x$ for some $i$ and $b_j = x - \alpha$ for every $j \neq i$. Then for every $k$-approximate mechanism we have that $f(\mathbf{b}) \in \left[x - \alpha\left(1 + \frac{k-1}{n}\right), x - \alpha\left(1 - \frac{k-1}{n-2}\right)\right]$.*

*Proof.* The optimal facility location for the given setting consists in placing the facility in position $x - \alpha$. The total cost in this case is $\alpha$.

If $f(\mathbf{b}) < x - \alpha\left(1 + \frac{k-1}{n}\right)$, then the total cost is larger than $(n-1)\frac{(k-1)\alpha}{n} + \alpha + \frac{(k-1)\alpha}{n} = k\alpha$, thus no $k$-approximate mechanism can place the facility in $f(\mathbf{b})$. Similarly, if $f(\mathbf{b}) > x - \alpha\left(1 - \frac{k-1}{n-2}\right)$, then the total cost is $(n-1)(f-x+\alpha)+x-f = (n-2)(f-x)+(n-1)\alpha > k\alpha$, thus no $k$-approximate mechanism can place the facility in $f(\mathbf{b})$. □

We are now ready to prove Theorem 3.

*Proof of Theorem 3.* Suppose there is an OSP mechanism $\mathcal{M}$ that is $(n-1-\varepsilon)$-approximate. Clearly, the mechanism is non-trivial, otherwise its approximation ratio would be unbounded. Then, let $i$ be the divergent agent of $\mathcal{M}$, and let $x_i$ and $y_i$ be the types in which $i$ diverges. W.l.o.g., assume that $x_i > y_i$. Let $\lambda = 2\left(x_i - y_i\right)$ and $\alpha = \lambda \cdot \frac{n-2}{\varepsilon}$. Let $x_i$ be the truthful position of this agent. If $i$ plays truthfully, then she can face the setting in which the remaining $n-1$ agents are in position $x_i - \alpha$. By applying Lemma 1 with $k = n-1-\varepsilon$ and $x = x_i$, we have that the distance of agent $i$ from the facility must be at least $x_i - x_i + \alpha\left(1 - \frac{n-2-\varepsilon}{n-2}\right) = \alpha \cdot \frac{\varepsilon}{n-2} = \lambda$.

Suppose that instead $i$ plays as if her real location would be $y_i$. It may be then the case that the remaining $n-1$ agents are exactly in the same position. Then, any mechanism with bounded approximation must place the facility in $y_i = x_i - \frac{\lambda}{2}$. Recall that, with monitoring, the cost of agent $i$ must be taken as the maximum between the distance to the facility either from the real position or from the declared position. In this case, this is given by the former distance and it is $\frac{\lambda}{2} < \lambda$. Thus, the best cost paid by $i$ by not playing truthfully is lower than the worst cost that she can pay by playing truthfully. Then, the mechanism $\mathcal{M}$ is not OSP, contradicting our hypothesis. □

The bound above is tight, as showed by the next theorem.

**Theorem 4.** *There is a $(n-1)$-approximate mechanism without money for the facility location problem that is OSP, even without monitoring.*

*Proof.* Consider the dictatorship mechanism, in which only the dictator $i$ is queried for her position. It is well-known that this mechanism is $(n-1)$-approximate. We next prove that it is also OSP. Agent $i$ is the only agent that is involved in a decision and it is always better for her to reveal her real position $x_i$: indeed, in this case the facility will be located exactly in her position and the cost of $i$ will be 0, whereas by declaring a different position $x \neq x_i$ the cost will be $|x - x_i| > 0$. □

## Facility Location with Money

**Theorem 5.** *For every $\varepsilon > 0$, there is no $\left(\frac{n}{2} - \varepsilon\right)$-approximate mechanism for the facility location problem that is OSP without monitoring.*

*Proof.* Let $\mathcal{M} = (f, \mathbf{p})$ be a $\left(\frac{n}{2} - \varepsilon\right)$-approximate mechanism that is OSP without monitoring. Let us restrict the domain of agent $i$ to $D_i' = \{a, b\}$, with $a < b$. Let $\mathcal{M}'$ be the mechanism obtained by pruning $\mathcal{M}$ according to this restriction. As stated above, $\mathcal{M}'$ must be an OSP mechanism. Moreover, the approximation ratio of $\mathcal{M}'$ cannot be worse than the approximation ratio of $\mathcal{M}$. Hence, $\mathcal{M}'$ cannot be trivial, otherwise its approximation ratio would be unbounded.

Then, let $i$ be the divergent agent of $\mathcal{M}'$. Clearly, $a$ and $b$ are the types in which $i$ diverges. Let $\mathbf{x}$ be the profile such that $x_i = b$ and $x_j = a$ for every $j \neq i$, and let $\mathbf{y}$ be the profile such that $y_i = a$ and $y_j = b$ for every $j \neq i$. Without loss of generality we can assume that

$$c_i(y_i, \mathcal{M}'(\mathbf{y})) \geq c_i(x_i, \mathcal{M}'(\mathbf{x})). \tag{1}$$

Lemma 1 with $x = b$, $\alpha = b - a$, $k = \frac{n}{2} - \varepsilon$ implies that $f(\mathbf{x}) \in \left[a - \alpha\left(\frac{1}{2} - \frac{1+\varepsilon}{n}\right), b - \alpha\left(\frac{1}{2} + \frac{\varepsilon}{n-2}\right)\right]$. Then, $d(x_i, f(\mathbf{x})) \geq \alpha\left(\frac{1}{2} + \frac{\varepsilon}{n-2}\right)$, and $d(y_i, f(\mathbf{x})) \leq \alpha \cdot \max\left\{\left(\frac{1}{2} - \frac{1+\varepsilon}{n}\right), \left(\frac{1}{2} - \frac{\varepsilon}{n-2}\right)\right\}$. Thus, $d(x_i, f(\mathbf{x})) > d(y_i, f(\mathbf{x}))$.

Since $\mathcal{M}'$ is OSP and $\mathbf{x}$ and $\mathbf{y}$ diverge, it must be the case that

$$\begin{aligned} c_i(y_i, \mathcal{M}'(\mathbf{y})) &\leq c_i(y_i, \mathcal{M}'(\mathbf{x})) = d(y_i, f(\mathbf{x})) + p_i(\mathbf{x}) \\ &< d(x_i, f(\mathbf{x})) + p_i(\mathbf{x}) = c_i(x_i, \mathcal{M}'(\mathbf{x})). \end{aligned}$$

The theorem then follows since the inequality above contradicts (1). □

Interestingly, monitoring gives an enormous power in this setting. We are going to assume that we are given some bounds on the agents' potential locations. (Note that in some of the related literature on facility location, agents can declare any location in $\mathbb{R}$.) To simplify the notation, we assume that $D_i = [a, b]$ for all agents $i$. Consider now the following direct-revelation mechanism, that we call *interval mechanism*:

1. Query agents for their position.
2. Let $\mathbf{x}$ be the profile of the collected positions. Then fix the location $f(\mathbf{x})$ of the facility to be the median of $\mathbf{x}$. In case of multiple medians, the facility is located on the leftmost median.
3. For every agent $i = 1, \ldots, n$, set $p_i(\mathbf{x}) = d(x_i, f(\mathbf{x})) - (b - a)$.

**Theorem 6.** *The interval mechanism is an optimal mechanism that is OSP with monitoring.*

*Proof.* We will next prove that the mechanism is OSP, and thus each agent has an incentive to declare her real position. Since the mechanism places the facility in the median of these positions, it then turns out to be optimal as well.

In order to prove that it is OSP, recall that in a mechanism with monitoring the cost that $i$ pays is $c_i(x_i, \mathcal{M}(\mathbf{y})) = \max\{d(x_i, f(\mathbf{y})), d(y_i, f(\mathbf{y}))\} - p_i(\mathbf{y})$. Consider then agent $i$ and let $x_i$ be her real position. If $i$ declares the real position, then her total cost will be $b - a$. If $i$ declares a different position $x_i'$, then there are two cases: if $\min_{\mathbf{x}'_{-i}} c_i(x_i, \mathcal{M}(\mathbf{x}'))$ is achieved in a profile $\mathbf{x}'_{-i}$ such that $f(\mathbf{x}') \neq x_i'$, then

$$\begin{aligned} c_i(x_i, \mathcal{M}(\mathbf{x}')) &= \max\{d(x_i, f(\mathbf{x}')), d(x_i', f(\mathbf{x}'))\} - p_i(\mathbf{x}') \\ &\geq d(x_i', f(\mathbf{x}')) - p_i(\mathbf{x}') = b - a; \end{aligned}$$

otherwise (that is, if $f(\mathbf{x}') = x_i' \neq x_i$)

$$\begin{aligned} c_i(x_i, \mathcal{M}(\mathbf{x}')) &= \max\{d(x_i, f(\mathbf{x}')), d(x_i', f(\mathbf{x}'))\} - p_i(\mathbf{x}') \\ &= d(x_i, x_i') - p_i(\mathbf{x}') > b - a. \end{aligned}$$

Thus, in both cases the best cost that $i$ can obtain by declaring a position different from the real one is not smaller than the worst cost that $i$ can obtain by playing truthfully. □

## Conclusions

We have studied the limitations of OSP mechanisms in terms of the approximation guarantee of their outputs. By focusing on two paradigmatic problems in the literature, machine scheduling and facility location, we have shown that OSP can yield a significant loss in the quality of the solutions returned. We have proposed the use of a novel mechanism design paradigm, namely monitoring, as a way to reconcile OSP with good approximations. Our positive results show how the ingredients needed for truthfulness with monitoring marry up the demands needed for OSP.

We leave open the problem of understanding the extent to which this parallel holds in general. Several additional open problems pertain the two case studies considered. For machine scheduling, it would be interesting to see whether the lower bound can be improved and/or understand how to deal with infinite domains. For facility location, the interval mechanism effectively charges the agents so that their utilities equal the length of the interval. Do less punitive mechanisms exist? Can we design OSP mechanisms for unbounded domains? More generally, the mechanisms with monitoring for which we provide an OSP implementation are shown to be collusion-resistant; does a concept of obvious collusion-resistance make sense? Would our mechanisms satisfy this notion?

**Acknowledgements.** Carmine Ventre was supported by the EPSRC grant EP/M018113/1.

## References

Adamczyk, M.; Borodin, A.; Ferraioli, D.; de Keijzer, B.; and Leonardi, S. 2015. Sequential posted price mechanisms with correlated valuations. In *International Conference on Web and Internet Economics*, 1–15. Springer.

Archer, A., and Tardos, É. 2001. Truthful mechanisms for one-parameter agents. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001*, 482–491.

Ashlagi, I., and Gonczarowski, Y. A. 2015. No stable matching mechanism is obviously strategy-proof. *arXiv preprint arXiv:1511.00452*.

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

Babaioff, M.; Immorlica, N.; Lucier, B.; and Weinberg, S. M. 2014. A simple and approximately optimal mechanism for an additive buyer. In *55th Annual Symposium on Foundations of Computer Science (FOCS), 2014*, 21–30. IEEE.

Brânzei, S., and Procaccia, A. D. 2015. Verifiably truthful mechanisms. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015*, 297–306.

Chawla, S.; Hartline, J. D.; Malec, D. L.; and Sivan, B. 2010. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the forty-second ACM symposium on Theory of computing*, 311–320. ACM.

Christodoulou, G., and Kovács, A. 2013. A deterministic truthful PTAS for scheduling related machines. *SIAM J. Comput.* 42(4):1572–1595.

Hochbaum, D. S., and Shmoys, D. B. 1988. A polynomial approximation scheme for scheduling on uniform processors: Using the dual approximation approach. *SIAM journal on computing* 17(3):539–551.

Koutsoupias, E., and Vidali, A. 2007. A lower bound of 1+ $\varphi$ for truthful scheduling mechanisms. In *International Symposium on Mathematical Foundations of Computer Science*, 454–464. Springer.

Kovács, A.; Meyer, U.; and Ventre, C. 2015. Mechanisms with monitoring for truthful ram allocation. In *International Conference on Web and Internet Economics*, 398–412. Springer.

Li, S. 2015. Obviously strategy-proof mechanisms. *Available at SSRN 2560028*.

Moulin, H. 1980. On strategy-proofness and single-peakedness. *Public Choice* 35:437–455.

Nisan, N., and Ronen, A. 2001. Algorithmic Mechanism Design. *Games and Economic Behavior* 35:166–196.

Penna, P., and Ventre, C. 2014. Optimal collusion-resistant mechanisms with verification. *Games and Economic Behavior* 86:491–509.

Sandholm, T., and Gilpin, A. 2003. Sequences of take-it-or-leave-it offers: Near-optimal auctions without full valuation revelation. In *International Workshop on Agent-Mediated Electronic Commerce*, 73–91. Springer.

Serafino, P.; Vidali, A.; and Ventre, C. 2016. Towards a characterization of budget-feasible mechanisms with monitoring.