Mechanism Design: (Ir)Rationality and Obvious Strategyproofness[⋆]

Diodato Ferraioli[1,*,†], Carmine Ventre[2,†]

[1] *Università degli Studi di Salerno, Fisciano SA 84084, Italy*
[2] *King's College London, London WC2R 2LS, UK*

Abstract

Multi-agent systems (MAS) are comprised by autonomous agents, each with a potentially specific goal that may be different from the objective of the system designer. MAS represent the perfect environment for the work in Algorithmic Mechanism Design (AMD), which seeks to design incentive-compatible mechanisms, the core idea being to maximise the profit of the agents *when* they behave honestly, thus preventing misbehaviour and allowing the designer to optimise her goal.

AMD often assumes full rationality of agents who are expected to know their full preferences (however complex they are) and to strategise optimally so that the mechanism is guided towards outcomes they prefer. However, in real MAS, this is too strong an assumption. Humans could interact with software agents and irrationally choose suboptimal strategies due to their cognitive biases and/or limitations [1]. Software agents themselves could be "irrational" since they could have been "badly" programmed either because the programmer misunderstood the incentive structure in place or due to computational barriers [2].

Much work has been done in the last years to relax full rationality and set an agenda to design AMD mechanisms for *real* MAS, where we seek to incentivise honest behaviour when agents have some form of imperfect rationality. This paper will survey some recent works focusing on mechanism design when agents have imperfect rationality.

Keywords

Mechanism Design, Bounded Rationality, Limited Contingent Reasoning, Greedy Algorithms

1. Introduction

Mechanism Design provides tools for developing protocols that align the goals of a planner with the selfish interests of the participating agents. Indeed, agents may, in principle, have an advantage if they deviate from the protocol's prescriptions. This could invalidate the guarantees of the protocol, such as, the maximization of some social measure of welfare or the revenue of the designer, that only hold under the assumption that agents behave as dictated. Hence, the goal is to design special protocols, named *mechanisms*, that allow to optimize the planner goals, and

22nd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2023) - Discussion Papers

[⋆] Based on [3, 4, 5, 6, 7].

[*] Corresponding author.

[†] These authors contributed equally.

dferraioli@unisa.it (D. Ferraioli); carmine.ventre@kcl.ac.uk (C. Ventre)
https://docenti.unisa.it/diodato.ferraioli (D. Ferraioli); https://www.kcl.ac.uk/people/carmine-ventre (C. Ventre)
0000-0002-7962-5200 (D. Ferraioli); 0000-0003-1464-1215 (C. Ventre)

at the same time incentivize agents to follow the protocol, a property called *strategyproofness*.

In Artificial Intelligence, mechanism design has found applications in many settings: from allocation, to facility location and matching problems [2].

Recently, a lot of interest has been devoted to designing mechanisms that not only aim to maximize the goal of the planner and to incentivize the correct behaviour of agents, but are also *simple*. Simplicity is usually intended in terms of the ability for the agents to understand their incentives without the need to engage in complex case analyses. From this point of view, simplicity is related with the transparency and the accountability of the protocol, that are often desirable properties, especially for democratic institutions.

This definition of simplicity has been recently formalized by Li [8] with the concept of *Obviously Strategyproof (OSP)* mechanisms. Roughly speaking, a mechanism is OSP if whenever it requires an agent to take an action, the worst outcome that she can achieve by following the protocol is not worse than the best outcome that she can achieve by deviating. Unfortunately, it has been observed that designing efficient OSP mechanisms can be a hard task [9], and indeed, most of the early works on the subject focus on special mechanism formats that are OSP, as posted price mechanisms [10, 11] and deferred acceptance auctions [12].

In this paper we describe a characterization of OSP mechanisms for single-parameter problems – wherein agent behaviour depends on a single parameter, also known as *type*. Interestingly, this characterizations relate OSP mechanisms to greedy and reverse greedy (a.k.a., deferred acceptance) algorithms, stating that algorithms with this format can be easily enriched with payments to guarantee obvious strategyproofness.

2. Notation

We let N denote a set of n *selfish agents* and $\mathcal{S}$ a set of feasible *outcomes*. Each agent i has a *type* $t_i \in D_i$ that we assume to be her *private knowledge*. We call D_i the *domain* of i. With $t_i(X) \in \mathbb{R}$ we denote the *cost* of agent i with type t_i for the outcome $X \in \mathcal{S}$. When costs are negative, the agent has a profit from the solution, called *valuation*. We will be working with costs and use that terminology accordingly but our results do not assume that costs are positive.

A *mechanism* interacts with the agents in N to select an outcome $X \in \mathcal{S}$. Specifically, agent i takes *actions* (e.g., saying yes/no) that may signal to the mechanism a type $b_i \in D_i$ different from t_i (e.g., saying yes could signal that the type has some properties that b_i has but t_i does not). We then say that agent i takes *actions compatible with (or according to)* b_i and call b_i the presumed type.

For a mechanism $\mathcal{M}$, $\mathcal{M}(\mathbf{b})$ denotes the outcome returned by the mechanism when the agents take actions according to their presumed types $\mathbf{b} = (b_1, \dots, b_n)$ (i.e., each agent i takes actions compatible with the corresponding b_i). This outcome is computed by a pair (f, p), where $f = f(\mathbf{b}) = (f_1(\mathbf{b}), \dots, f_n(\mathbf{b}))$ (termed *social choice function* or algorithm) maps the actions taken by the agents according to $\mathbf{b}$ to a feasible solution in $\mathcal{S}$, and $p(\mathbf{b}) = (p_1(\mathbf{b}), \dots, p_n(\mathbf{b})) \in \mathbb{R}^n$ maps the actions taken by the agents according to $\mathbf{b}$ to *payments*.

Each selfish agent is equipped with a *quasi-linear utility function*, i.e., agent i has utility function $u_i \colon D_i \times \mathcal{S} \to \mathbb{R}$: for $t_i \in D_i$ and for an outcome $X \in \mathcal{S}$ returned by a mechanism $\mathcal{M}$, $u_i(t_i, X)$ is the utility that agent i has for the implementation of outcome X when her type is t_i, i.e., $u_i(t_i, \mathcal{M}(b_i, \mathbf{b}_{-i})) = p_i(b_i, \mathbf{b}_{-i}) - t_i(f(b_i, \mathbf{b}_{-i}))$.

A *single-parameter* agent i has as private information a single real number t_i and $t_i(X)$ can be expressed as $t_i\mathsf{w}_i(X)$ for some publicly known function w; note that $\mathsf{w}_i(X)$ is a non-negative real number (and $\mathcal{S} = \mathbb{R}^n_{\geq 0}$). Moreover, observe that the cost of player i is independent on what the outcome X prescribes for players different from i. We make no other assumption on $\mathcal{S}$. To simplify the notation, we will write $t_i f_i(\mathbf{b})$ when we want to express the cost of a single-parameter agent i of type t_i for the output of social choice function f on input the actions corresponding to a bid vector $\mathbf{b}$.

3. Obvious Strategyproofness

We here introduce the concept of implementation tree to formally define (deterministic) OSP mechanisms. Our definition is, w.l.o.g., built on the one by Mackenzie [13] rather than the original definition by Li [8].

An *extensive-form mechanism* $\mathcal{M}$ is a triple $(f, p, \mathcal{T})$ where, as from above, the pair (f, p) determines the outcome of the mechanism, and $\mathcal{T}$ is a tree, called *implementation tree*, such that:

- Every leaf ℓ of the tree is labeled with a possible outcome of the mechanism $(X(\ell), p(\ell))$, where $X(\ell) \in \mathcal{S}$ and $p(\ell) \in \mathbb{R}$;
- Each node u in the implementation tree $\mathcal{T}$ defines the following:
  - An agent $i = i(u)$ to whom the mechanism makes a query. Each possible answer to this query leads to a different child of u.
  - A subdomain $D^{(u)} = (D_i^{(u)}, D_{-i}^{(u)})$ containing all types that are *compatible* with u, i.e., compatible with all the answers to the queries from the root down to node u. Specifically, the query at node u defines a partition of the current domain of $i = i(u)$, $D_i^{(u)}$ into $k \geq 2$ subdomains, one for each of the k children of node u. Thus, the domain of each of these children will have as the domain of i, the subdomain of $D_i^{(u)}$ corresponding to a different answer of i at u, and an unchanged domain for the other agents.

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 $D^{(\ell)}$. Similarly, to each leaf ℓ there is at least a profile $\mathbf{b}$ that belongs to $D^{(\ell)}$. For this reason, we say that $\mathcal{M}(\mathbf{b}) = (X(\ell), p(\ell))$.

Two profiles $\mathbf{b}, \mathbf{b}'$ are said to *diverge* at a node u of $\mathcal{T}$ if this node has two children v, v' such that $\mathbf{b} \in D^{(v)}$, whereas $\mathbf{b}' \in D^{(v')}$. For every such node u, we say that $i(u)$ is the *divergent agent* at u.

We are now ready to define obvious strategyproofness. An extensive-form mechanism $\mathcal{M}$ is *obviously strategy-proof (OSP)* if for every agent i with real type t_i, for every vertex u such that $i = i(u)$, 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 $u_i(t_i, \mathcal{M}(t_i, \mathbf{b}_{-i})) \geq u_i(t_i, \mathcal{M}(b_i, \mathbf{b}'_{-i}))$. Roughly speaking, an OSP mechanism requires that, at each time step agent i is asked to take a decision that depends on her type, the worst utility that she can get if she behaves according to her true type is at least the best utility she can get by behaving differently.

4. Cycle-monotonicity Characterizes OSP Mechanisms

We next describe the main tools needed for our characterization: i.e., OSP can be characterized by the absence of negative-weight cycles in a suitable weighted graph over the possible strategy profiles. Specifically, we consider a mechanism $\mathcal{M}$ with implementation tree $\mathcal{T}$ for a social choice function f, and define:

- **Separating Node:** A node u in the implementation tree $\mathcal{T}$ is $(\mathbf{a}, \mathbf{b})$-separating for agent $i = i(u)$ if $\mathbf{a}$ and $\mathbf{b}$ are compatible with u (that is, $\mathbf{a}, \mathbf{b} \in D^{(u)}$), and the two types a_i and b_i belong to two different subdomains of the children of u (thus implying $a_i \neq b_i$).
- **OSP-graph:** For every agent i, we define a directed weighted graph $\mathcal{O}_i^{\mathcal{T}}$ having a node for each profile in $D = \times_i D_i$. The graph contains edge $(\mathbf{a}, \mathbf{b})$ if and only if $\mathcal{T}$ has some node u which is $(\mathbf{a}, \mathbf{b})$-separating for $i = i(u)$, and the weight of this edge is $w(\mathbf{a}, \mathbf{b}) = a_i(f_i(\mathbf{b}) - f_i(\mathbf{a}))$. Throughout the paper, we will denote with $\mathbf{a} \to \mathbf{b}$ an edge $(\mathbf{a}, \mathbf{b}) \in \mathcal{O}_i^{\mathcal{T}}$, and with $\mathbf{a} \rightsquigarrow \mathbf{b}$ a path among these two profiles in $\mathcal{O}_i^{\mathcal{T}}$.
- **OSP Cycle Monotonicity (OSP CMON):** OSP cycle monotonicity (OSP CMON) holds if, for all i, the graph $\mathcal{O}_i^{\mathcal{T}}$ does not contain negative-weight cycles. Moreover, OSP two-cycle monotonicity (OSP 2CMON) holds if the same is true when considering cycles of length two only, i.e., cycles with only two edges. Sometimes, we will simply say CMON and 2CMON below.

Theorem 1. *A mechanism with implementation tree $\mathcal{T}$ for a social function f is OSP on finite domains if and only if OSP CMON holds. Moreover, for any OSP mechanism $\mathcal{M} = (f, p, \mathcal{T})$ where $\mathcal{T}$ is not a binary tree, there is an OSP mechanism $\mathcal{M}' = (f, p, \mathcal{T}')$ where $\mathcal{T}'$ is a binary tree.*

Given the result above, we henceforth assume that the agents have finite domains and that the implementation trees of our mechanisms are binary.

5. Algorithmic Characterization of OSP Mechanisms

We first observe that it is w.l.o.g. to restrict to a specific class of mechanisms, that are ordered. Specifically, let $\mathcal{M} = (f, p, \mathcal{T})$ be an extensive-form mechanism. Let $u \in \mathcal{T}$ be a node where $i = i(u)$ and $D_i^{(u)}$ is separated into L and R. We say that the query at u is *ordered* if for all $l, r \in D_i^{(u)}$ with l in L and r in R, $l < r$ and $\mathcal{L}_i^{(u)}(r) \preceq \mathcal{L}_i^{(u)}(l)$. Then, we say that a mechanism is ordered if it only makes ordered queries. Next result shows that we can focus on ordered mechanisms without loss of generality as long as we are interested in OSP.

Theorem 2. *Any OSP mechanism $\mathcal{M} = (f, p, \mathcal{T})$ can be transformed into an equivalent OSP mechanism $\mathcal{M}' = (f, p, \mathcal{T}')$ where all queries in $\mathcal{T}'$ are ordered.*

In order to provide our algorithmic characterization, we begin by defining the concept of antimonotone types and of pivots for a pair of types. We say that two types like $b_i^{(1)} > b_i^{(2)}$ for which there are profiles $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$ such that $f_i(\mathbf{b}^{(1)}) > f_i(\mathbf{b}^{(2)})$ are *antimonotone* and call the profiles $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$ *witnesses* of antimonotonicity of $b_i^{(1)}$ and $b_i^{(2)}$. Given a node u and a pair of types $b_i^{(1)}, b_i^{(2)} \in D_i^{(u)}$, we say that types $b_i^{(u)}$, $b_i^{(d)}$ are *pivots* for $b_i^{(1)}$ and $b_i^{(2)}$ if

- they are separated from $b_i^{(1)}, b_i^{(2)}$ respectively at nodes $v^{(u)}, v^{(d)} \in \mathcal{T}$ that are ancestors of u with $i(v^{(u)}) = i(v^{(d)}) = i$;

- for $y \in \mathcal{L}_i^{(u)}(b_i^{(1)})$ and $x \in \mathcal{L}_i^{(u)}(b_i^{(2)})$ with $y > x$, there are $z \in \mathcal{L}_i^{(v^{(u)})}(b_i^{(u)})$ with $z \geq y$, and $q \in \mathcal{L}_i^{(v^{(d)})}(b_i^{(d)})$ with $q \leq x$.

For a label $x \in \mathcal{L}^{(u)}(t)$ of some type t at node $u \in \mathcal{T}$, we call $\mathbf{b}_{-i}$ a x-buddy for t at u if $\mathbf{b}_{-i} \in D_{-i}^{(u)}$ and $f_i(t, \mathbf{b}_{-i}) = x$. Given a node u and a pair of types $b_i^{(1)}, b_i^{(2)} \in D_i^{(u)}$, we say that types $b_i^{(u)}$, $b_i^{(d)}$ are *extreme* pivots if they are pivots for $b_i^{(1)}$ and $b_i^{(2)}$ and, given $z \geq y > x \geq q$ as above, we have that $w(P_1^{(out)}) + w(P_2^{(in)}) + w(P_2^{(out)}) + w(P_1^{(in)}) \geq 0$, for all paths $P_1^{(out)}$, $P_2^{(in)}$, $P_2^{(out)}$, $P_1^{(in)}$ in $\mathcal{O}_i^{\mathcal{T}}$ defined as follows:

$$\begin{aligned}
P_1^{(out)} &:= (b_i^{(1)}, \mathbf{b}_{-i}^{(y)}) \to \mathbf{a}^{(1)} \rightsquigarrow \mathbf{a}^{(k)} \to (b_i^{(d)}, \mathbf{b}_{-i}^{(q)}) \\
P_2^{(in)} &:= (b_i^{(d)}, \mathbf{b}_{-i}^{(q)}) \to \mathbf{c}^{(1)} \rightsquigarrow \mathbf{c}^{(\ell)} \to (b_i^{(2)}, \mathbf{b}_{-i}^{(x)}) \\
P_2^{(out)} &:= (b_i^{(2)}, \mathbf{b}_{-i}^{(x)}) \to \mathbf{c}^{(\ell+1)} \rightsquigarrow \mathbf{c}^{(\ell+h)} \to (b_i^{(u)}, \mathbf{b}_{-i}^{(z)}) \\
P_1^{(in)} &:= (b_i^{(u)}, \mathbf{b}_{-i}^{(z)}) \to \mathbf{a}^{(k+1)} \rightsquigarrow \mathbf{a}^{(k+g)} \to (b_i^{(1)}, \mathbf{b}_{-i}^{(y)}),
\end{aligned}$$

where $\mathbf{b}_{-i}^{(y)}$ is a y-buddy for $b_i^{(1)}$ at u, $\mathbf{b}_{-i}^{(x)}$ is an x-buddy for $b_i^{(2)}$ at u, $\mathbf{b}_{-i}^{(q)}$ is a z-buddy for $b_i^{(d)}$ at $v^{(d)}$, $\mathbf{b}_{-i}^{(z)}$ is a q-buddy for $b_i^{(u)}$ at $v^{(u)}$, $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(k+g)}$ and $\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(\ell+h)}$ are profiles in $\mathcal{O}_i^{\mathcal{T}}$.

We are now ready to provide the definition of the mechanism format that characterizes OSP: A mechanism $\mathcal{M} = (f, p, \mathcal{T})$ is *three-way greedy* if all its queries are ordered and for all internal nodes $u \in \mathcal{T}$ such that $i(u) \neq i$ and $b_i^{(1)}$ and $b_i^{(2)}$ in $D_i^{(u)}$ are antimonotone, it holds that any pair of pivots $b_i^{(u)}$ and $b_i^{(d)}$ are extreme. We then have the following theorem:

Theorem 3. *An OSP mechanism $\mathcal{M}$ implementing f exists if and only if a three-way greedy mechanism implementing f exists.*

To make sense of the notion (and the name) of three-way greedy mechanisms, we now explore few of their properties. Let us first assume that we would like to avoid the introduction of pivots until there are antimonotone types (this will surely satisfy the definition of three-way greedy mechanism). How can a mechanism avoid two pivots? Clearly, the mechanism can query an agent in a greedy fashion (i.e., by querying about the best type that has not yet been queried, and in case of positive answer, by guaranteeing her an outcome at least as good as the one she may achieve in case of negative answer) or in a reverse greedy fashion (i.e., by asking her whether her type is the worst that has not yet been queried, and in case of positive answer, by guaranteeing her an outcome at least as bas as the one she may achieve in case of positive answer). The third possibility is for the mechanism to first query an agent about whether her type is large or small (with the exact threshold defining large or small types depending on the problem at the hand), whilst ensuring that a label for a large type is never better than the label of a small type, and then in case of large types, proceeding by querying the agent greedily, whereas in the case of small types, the mechanism queries the agent in a reverse greedy fashion. These three ways of ensuring that no two pivots exist justify the name of our mechanism. Actually, the definition of three-way greedy mechanisms allows pivots to exist, as long as the one with small (large) label is large (small) *enough*. Here, the thresholds for these pivots to be considered small/large enough depend on cycles that go through the four aforementioned points, cf. the definitions of the paths in the OSP-graph.

Interestingly, this leads to an even simpler characterization in case of binary allocation problems. Indeed, with only two outcomes available, the only pivots possible must have outcomes that are equal to those of the two antimonotone types. This implies that the existence of pivots leads to negative-weight cycles. Hence, the only way to satisfy the definition of three-way greedy mechanism is to avoid pivots, that in turn means that the mechanism has to interact with each agent either in a greedy fashion or in a reverse greedy fashion as long as there are still antimonotone types (for this reason, we term such a mechanism *two-way greedy* mechanisms).

6. Payments

Theorem 3 is essentially existential, since it does not provide explicit payments. The existence of the payments follows from Theorem 1: the payments for a particular player are defined therein as the shortest path in the corresponding OSP graph. However, these graphs have in general exponential size with respect to the description of the instance, meaning that this approach is infeasible from a computational point of view. Moreover, the implicit definition of payments "hides" the simplicity of the decision making of agents facing an OSP mechanism. We next show that these payments actually have a simple structure.

To this aim, let $\mathcal{M}$ be a mechanism with a three-way greedy implementation for a social function f. We say that the outcomes corresponding to bid profiles $\mathbf{a}$ and $\mathbf{b}$ are *equivalent* to agent i, denoted as $\mathbf{a} =_i \mathbf{b}$, whenever $f_i(\mathbf{a}) = f_i(\mathbf{b})$, and that agent i prefers X to Y, denoted as $X \succ_i Y$, whenever $f_i(\mathbf{a}) > f_i(\mathbf{b})$. Hence, we can partition profile types in equivalence classes $X_i^0, \dots, X_i^m$, for some $m \geq 0$ such that $X_i^0 = \{\mathbf{b}\colon f_i(\mathbf{b}) = \min_{\mathbf{a}} f_i(\mathbf{a})\}$, i.e., it contains all bid profiles returning the minimum outcome to i, and $X_i^j = \{\mathbf{b}\colon f_i(\mathbf{b}) = \min_{\mathbf{a} \notin X_i^0, \dots, X_i^{j-1}} f_i(\mathbf{a})\}$, i.e. it contains all bid profiles returning to i the smallest outcome larger than the one returned by profiles in previous equivalence classes. We also define $X_i^{<j} = \bigcup_{\ell=0}^{j-1} X_i^\ell$ and $X_i^{>j} = \bigcup_{\ell=j+1}^{m} X_i^\ell$. Moreover for $j = 1, \dots, m$, we also let $f_i^j = f_i(\mathbf{b})$ for some $\mathbf{b} \in X_i^j$. Finally, given a profile $\mathbf{b}'$ we will say that it is *related to* $\mathbf{b}$ if $\mathbf{b}$ and $\mathbf{b}'$ are either not separated until agent i is queried about type b_i, or they have been separated by i. Now, for $j = 0, \dots, m$, and every $\mathbf{b}$ let $\theta_{\mathbf{b}}(j) = \max_{\substack{\mathbf{b}' \in X_i^j \\ \mathbf{b}' \text{ related to } \mathbf{b}}} b_i'$. That is, $\theta_{\mathbf{b}}(j)$ is the largest bid which may cause the assignment of outcome f_i^j to agent i on the path from the root of $\mathcal{T}$ until agent i is queried about b_i.

We will start by defining the payment for an agent i that interacts with this mechanism in a reverse greedy fashion (i.e., the agent is queried for the worst type not yet queried, and upon a positive answer she receives an outcome not larger than the outcome received by declaring a better type).

Proposition 4. *Let $\mathcal{M}$ be a mechanism with a three-way greedy implementation and let i be an agent interacting with $\mathcal{M}$ in a reverse greedy way. Then truthfulness is an obvious dominant strategy for i if for every $\mathbf{b} \in X_i^k$ $p_i(\mathbf{b}) = \theta_{\mathbf{b}}(k) f_i^k + \sum_{j=0}^{k-1} (\theta_{\mathbf{b}}(j) - \theta_{\mathbf{b}}(j+1)) f_i^j$.*

It is immediate to check that payments defined in Proposition 4 are essentially the same as strategyproof payments as defined in [14].

Let us now consider an agent i that interacts with the mechanism in a greedy fashion (i.e., the agent is queried for the best type not yet queried, and upon a positive answer she receives an outcome that is not smaller than the outcome received by declaring a worse type). To this aim we let $q(j)$, for $j = 1, \dots, m$, be the type corresponding to the first query in the tree that, if positively answered, will assign to agent i the outcome f_i^j, that is the smallest type on which a query is issued with promised outcome f_i^j. Moreover, we let $\tau(0) = \max_{\mathbf{b}} b_i$, and, for $j = 1, \dots, m$, $\tau(j) = \min_{\substack{\mathbf{b} \in X_i^{<j} \\ b_i > q(j)}} b_i$. That is, $\tau(j)$ is the smallest bid which may cause the assignment of an outcome worse than f_i^j to agent i after the query $q(j)$. Observe that for each b_i such that there is $\mathbf{b}_{-i}$ such that $(b_i, \mathbf{b}_{-1}) \in X_i^k$ we have that $\tau(k) \leq b_i$. We next show that payments in this case have a very similar structure as the one described above, but they fail to match SP payments.

Proposition 5. *Let $\mathcal{M}$ be a mechanism with a three-way greedy implementation tree and let i be an agent interacting with $\mathcal{M}$ in a greedy way. Let $X_i^0, \dots, X_i^m$ be the partition of type profiles in equivalence class for i as defined above. Then truthfulness is an obvious dominant strategy for i if for every $\mathbf{b} \in X_i^k$*

$$p_i(\mathbf{b}) = \begin{cases} \min \left\{ 0, \min_{\substack{b'_i \leq b_i \\ \exists \mathbf{b}'_{-i} \colon \mathbf{b}' \in X_i^{>0}}} (p_i(\mathbf{b}') - b'_i f_i(\mathbf{b}')) \right\} & \textit{if } k = 0; \\ \tau(k) f_i^k + \sum_{j=0}^{k-1} (\tau(j) - \tau(j+1)) f_i^j & \textit{o.w.}. \end{cases}$$

Note that there are two main differences between payments as defined in Proposition 5 and payments provided in Proposition 4: first, we changed the threshold for outcome f_i^j from the SP threshold $\theta(k)$ to a smaller threshold $\tau(j)$; second, the payment associated with the lowest outcome depends not only on the outcome, but also on when this outcome is achieved.

The third way our mechanism can interact with an agent consists in first asking to separate the domain in good and bad types (with outcomes for good types being not worse than outcomes for bad types), and then proceeding greedily over bad types and reverse greedily over good types. Hence, it is not surprising that in this case payments are a composition of the ones described above.

7. Applications

Our algorithmic characterization of OSP mechanisms, showing a connection with a certain family of greedy algorithms, allows us to show quite easily the existence of a host of new mechanisms, and to provide a set of upper bounds on their approximation guarantee. We summarize some of these results in Table 1.

Problem	Bound
Known Single-Minded Combinatorial Auctions (CAs)	$\sqrt{m}$
MST (& weighted matroids)	1
Max Weighted Matching	2
p-systems	p
Weighted Vertex Cover	2
Shortest Path	$\Omega(n)$
Restricted Knapsack Auctions	$\Omega(\sqrt{n})$
Asymmetric Restricted Knapsack Auctions (3 values)	$\sqrt{n-1}$
Knapsack Auctions	$\Omega(\sqrt{\ln n})$
Related Machine Scheduling	n
Related Machine Scheduling (4 speeds only)	$\frac{n}{2}+1$
Related Machine Scheduling (3 speeds only)	$\lceil\sqrt{n}\rceil+1$
Related Machine Scheduling (2 speeds only)	1

Table 1
Bounds on the approximation guarantee of OSP mechanisms. (A p-system is a downward-closed set system $(E, \mathcal{F})$ where there are at most p *circuits*, that is, minimal subsets of E not belonging to $\mathcal{F}$ [15].)

8. Conclusions

We believe that our characterization helps in the construction of simple incentive-compatible mechanisms for real MAS, in which agents may have imperfect rationality. Clearly, there are many more settings than the one showed in Table 1 in which it would be interesting to design these mechanisms, or to evaluate their approximation guarantee.

Moreover, we believe that our approach can be extended to work also with other definitions of simple mechanisms based on extensive form mechanisms, as Expected OSP [16], OSP with lookahead [17], k-step OSP [18], Non-Obvious Manipulations [19, 20].

References

[1] L. M. Ausubel, An efficient ascending-bid auction for multiple objects, American Economic Review 94 (2004) 1452–1475.

[2] N. Nisan, T. Roughgarden, E. Tardos, V. Vazirani (Eds.), Algorithmic Game Theory, 2017.

[3] D. Ferraioli, C. Ventre, Approximation guarantee of OSP mechanisms: The case of machine scheduling and facility location, Algorithmica 83 (2021) 695–725. URL: https://doi.org/10.1007/s00453-020-00771-x. doi:10.1007/s00453-020-00771-x.

[4] D. Ferraioli, A. Meier, P. Penna, C. Ventre, New constructions of obviously strategyproof mechanisms, Mathematics of Operations Research (2022).

[5] D. Ferraioli, P. Penna, C. Ventre, Two-way greedy: Algorithms for imperfect rationality, in: WINE, volume 13112 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 3–21.

[6] D. Ferraioli, C. Ventre, Explicit payments for obviously strategyproof mechanisms, in: AAMAS, ACM, 2023, pp. 2125–2133.

[7] D. Ferraioli, C. Ventre, On the connection between greedy algorithms and imperfect rationality, in: EC, ACM, 2023, pp. 657–677.

[8] S. Li, Obviously strategy-proof mechanisms, American Economic Review 107 (2017) 3257–87.

[9] D. Ferraioli, C. Ventre, Obvious strategyproofness needs monitoring for good approximations, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[10] M. Babaioff, N. Immorlica, B. Lucier, S. M. Weinberg, A simple and approximately optimal mechanism for an additive buyer, in: FOCS 2014, 2014, pp. 21–30.

[11] M. Adamczyk, A. Borodin, D. Ferraioli, B. de Keijzer, S. Leonardi, Sequential posted price mechanisms with correlated valuations, in: WINE 2015, 2015, pp. 1–15.

[12] P. Milgrom, I. Segal, Clock auctions and radio spectrum reallocation, Journal of Political Economy (2020).

[13] A. Mackenzie, A revelation principle for obviously strategy-proof implementation, Research Memorandum 014, Maastricht University, Graduate School of Business and Economics (GSBE), 2018.

[14] A. Archer, É. Tardos, Truthful mechanisms for one-parameter agents, in: 42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA, IEEE Computer Society, 2001, pp. 482–491. URL: https://doi.org/10.1109/SFCS.2001.959924. doi:10.1109/SFCS.2001.959924.

[15] D. Hausmann, B. Korte, T. A. Jenkyns, Worst case analysis of greedy type algorithms for independence systems, Combinatorial Optimization (1980) 120–131.

[16] D. Ferraioli, C. Ventre, Probabilistic verification for obviously strategyproof mechanisms, in: J. Lang (Ed.), Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, ijcai.org, 2018, pp. 240–246. URL: https://doi.org/10.24963/ijcai.2018/33. doi:10.24963/ijcai.2018/33.

[17] D. Ferraioli, C. Ventre, Obvious strategyproofness, bounded rationality and approximation, Theory Comput. Syst. 66 (2022) 696–720. URL: https://doi.org/10.1007/s00224-022-10071-2. doi:10.1007/s00224-022-10071-2.

[18] P. Troyan, T. Morrill, Obvious manipulations, Journal of Economic Theory 185 (2020) 104970.

[19] T. Archbold, B. De Keijzer, C. Ventre, Non-obvious manipulability for single-parameter agents and bilateral trade, in: Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), 2023.

[20] T. Archbold, B. De Keijzer, C. Ventre, Non-obvious manipulability in extensive-form mechanisms: the revelation principle for single-parameter agents, in: Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI 2023), 2023.