Explicit Payments for Obviously Strategyproof Mechanisms

Diodato Ferraioli
Università degli Studi di Salerno
Fisciano (SA), Italy
dferraioli@unisa.it

Carmine Ventre
King's College London
London, United Kingdom
carmine.ventre@kcl.ac.uk

ABSTRACT

The design of mechanisms where incentives are simple to understand for the agents has attracted a lot of attention recently. One particularly relevant concept in this direction has been Obvious Strategyproofness (OSP), a class of mechanisms that are so simple to be recognized as incentive compatible even by agents with a limited form of rationality. It is known that there exist payments that lead to an OSP mechanism whenever the algorithm they augment is either greedy or reverse greedy (a.k.a., deferred acceptance). However, to date, their explicit definition is unknown.

In this work we provide payments for OSP mechanisms based on greedy or reverse greedy algorithms. Interestingly, our results show an asymmetry between these two classes of algorithms: while for reverse greedy the usual strategyproof payments work well also for OSP, the payments for greedy algorithms may break individual rationality or budget balancedness. Thus, the designer needs to subsidize the market in order to simultaneously guarantee these properties and simple incentives. We apply this result to analyze the amount of subsidies needed by a well-known greedy algorithm for combinatorial auctions with single-minded bidders.

CCS CONCEPTS

• **Theory of computation → Algorithmic mechanism design**.

KEYWORDS

Bounded Rationality; Greedy Algorithms; Single-Minded Auctions

ACM Reference Format:
Diodato Ferraioli and Carmine Ventre. 2023. Explicit Payments for Obviously Strategyproof Mechanisms. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

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 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 [23].

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.

The vague definition of simplicity that we described above has been recently formalized by Li [20] 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, since the introduction of this concept, it has been observed that designing efficient OSP mechanisms can be a hard task [15], and indeed, most of the early work on the topic focuses on special mechanism formats observed to be OSP, such as, posted price mechanisms [1, 6] and deferred acceptance auctions [22].

Only recently, a characterization of OSP mechanisms has been provided in [14] for single-parameter problems – wherein agent behaviour depends on a single parameter, also known as *type* – and binary outcomes (i.e., where each agent either wins or loses) and very recently extended to general outcome spaces in [17]. Interestingly, both these characterizations relate OSP mechanisms to greedy and reverse greedy (a.k.a., deferred acceptance) algorithms, stating that algorithms with this format can be enriched with payments to guarantee obvious strategyproofness.

Unfortunately, the definition of these payments has not been explicitly provided in previous work. Clearly, this challenges the "simplicity" of these mechanisms based on greedy algorithms: what if the payments making them OSP are complex or hard to compute?

Our Contribution. In this work we provide explicit payments for OSP mechanisms based on greedy and reverse greedy algorithms. Specifically, it turns out that payments in the case of reverse greedy algorithms are essentially the same as the well-known payments for standard strategyproof mechanisms for single-parameter agents [2]. This, in turn, implies that this kind of implementation enjoys other desirable properties of mechanisms, such as individual rationality and budget balance (see below for formal definitions).

Interestingly, we observe an asymmetry with respect to the mechanisms built on greedy algorithms. We show that in this case, payments are different from the strategyproof ones, and in general they are not able to guarantee both individual rationality and balanced balancedness. That is, these payments may require one agent to be paid less that the cost they incur into by participating

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

to the mechanism. Hence, in order to incentivize the participation to the mechanism, the planner has to provide subsidies to agents, meaning that the planner will spend more than the actual cost of the solution computed by the mechanism.

This dichotomy addresses a problem raised in [22] about the OSPness of greedy algorithms. Indeed, it was observed therein that greedy mechanisms, when equipped with strategyproof payments, cease to be OSP, and this result was used to justify a preference of reverse greedy mechanism over greedy mechanisms. We can now better motivate this preference: reverse greedy mechanisms are indeed the only one to guarantee at the same time obvious strategyproofness, individual rationality, and budget balancedness.

Note that we provide our characterization of payments both in the case of binary outcomes (in Section 4) and general single parameter mechanisms (in Section 3).

We finally complete this work, by providing a specific case study for combinatorial auctions with single-minded bidders. In particular, we bound the amount of subsidies necessary to make OSP the well-known greedy algorithm of Lehmann et al. [19] for this setting. We prove that there are instance whose subsidies can be very large (essentially we have to pay each agent her second highest possible type), and instances in which no subsidies are necessary.

Other Related Work. OSP mechanisms attracted a lot of attention. Some works provide preliminary characterizations for these mechanisms. Specifically, some papers [7, 21, 25] aim at simplifying the notion of OSP, by looking at versions of the revelation principle for OSP mechanisms. This, for example, allows to focus, without loss of generality, on deterministic (rather than randomized) extensive-form mechanisms where each agent moves sequentially (rather than concurrently). More relevant to our paper is a technique to characterize OSP via *cycle-monotonicity*, defined in [13].

A few OSP mechanisms have been recently proposed. Most of them focus on restricted preference spaces, such as single-peaked domains [3, 4, 7] whereas others focus more on specific applications, e.g., OSP stable matching is studied in [5], public project selection in [16], machine scheduling in [13] and binary allocation problems in [12]. Negative results, such as inapproximability or impossibility results, about the performances of OSP mechanisms are similarly quite sparse. Some inapproximability results have been instead provided for special mechanisms formats, that can be observed to be OSP, such as deferred acceptance auctions. For example, [10] prove that the approximation guarantee of these mechanisms are quite poor compared to what strategyproof mechanisms can do for several optimization problems. Follow-up work is discussed in [8, 11]. However, it is not known whether these results extend to any OSP mechanism.

2 DEFINITIONS AND PRELIMINARIES

Consider a set N of n *selfish agents* (a.k.a., bidders) and a set $\mathcal{S}$ of feasible *outcomes*. Agent $i \in N$ has a *type* $t_i \in D_i$, D_i being the *domain* of i. We assume that the type of each agent is *private knowledge*. 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, it means that the agent has a so-called *valuation* for the solution. In the former case, we can think at the agent performing some work for the mechanism (á la procurement auction) whereas in the latter agents have a profit from the outcome computed by the mechanism (as in, e.g., single-item auctions). In the remainder of this paper, we will mainly be working with costs and use that terminology accordingly but none of our structural results on the payment functions assume that costs are positive. In fact, we will apply our results to a setting with valuations.

In general, a *mechanism* is a protocol between the designer and the agents in N; at the end of this interaction, the mechanism selects an outcome $X \in \mathcal{S}$. During the protocol, 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 a b_i which is smaller for a certain solution than t_i). In such a case, we say that agent i takes *actions compatible with (or according to)* b_i. We call b_i the presumed type of agent i. For a mechanism M, $M(\mathbf{b})$ denotes the outcome returned by the mechanism when the agents take actions according to their presumed types $\mathbf{b} = (b_1, \ldots, b_n)$ (i.e., each agent i takes actions compatible with the corresponding b_i). This outcome is computed by a pair (f, p), where $f = f(\mathbf{b}) = (f_1(\mathbf{b}), \ldots, f_n(\mathbf{b}))$ (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 $p(\mathbf{b}) = (p_1(\mathbf{b}), \ldots, p_n(\mathbf{b})) \in \mathbb{R}^n$ maps the actions taken by the agents according to $\mathbf{b}$ to *payments*. Note that payments need not be positive.

As common in the literature, we assume that each selfish agent has 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 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.,

$$u_i(t_i, M(b_i, \mathbf{b}_{-i})) = p_i(b_i, \mathbf{b}_{-i}) - t_i(f(b_i, \mathbf{b}_{-i})).$$

In this work, we focus on *single-parameter* agents. The private information of each single-parameter bidder i is a single real number t_i and $t_i(X)$ can be expressed as $t_i \mathrm{w}_i(X)$ for some function $\mathrm{w} : \mathcal{S} \to \mathbb{R}^n$ that is publicly known. Note that the cost of player i for outcome X is independent of $\mathrm{w}_j(X)$ for $j \neq i$. 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 $\mathbf{b}$. The binary outcome setup, wherein $f_i(\mathbf{b}) \in \{0, 1\}$, is a special case of particular interest. We will typically say that agent i wins at $\mathbf{b}$ (loses, respectively) whenever $f_i(\mathbf{b}) = 1$ ($f_i(\mathbf{b}) = 0$, respectively) omitting the dependence from $\mathbf{b}$ whenever this is clear from the context.

Extensive-form Mechanisms and Obvious Strategyproofness. We build our definition upon the one in [14], which is shown to be equivalent to the original notion in [20], and some structural results therein. We kindly refer the interested reader to [14] for details.

We begin with *extensive-form mechanisms*. An extensive-form mechanism 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 binary 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 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, $D_i^{(u)}$ into two subdomains, one for each of the 2 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.

For the implementation tree to be well defined, we shall assume that agents have finite domains, i.e., $|D_i|$ is a finite number.

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 $D^{(\ell)}$. We then say that $M(\mathbf{b}) = (X(\ell), p(\ell))$.

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 say that agent i with real type t_i adopts the truthtelling strategy for M if for every vertex u such that $i = i(u)$ and $t_i \in D_i^{(u)}$, agent i plays according to t_i at u. This truthtelling strategy of agent i is obviously dominant if 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,

$$u_i(t_i, M(t_i, \mathbf{b}_{-i})) \geq u_i(t_i, M(b_i, \mathbf{b}'_{-i})).$$

An extensive-form mechanism M is *obviously strategy-proof (OSP)* if the truthtelling strategy is obviously dominant for every agent i. In words, an obviously strategy-proof 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. For comparison, a mechanism M is strategyproof is truthtelling is a dominant strategy for each agent i, that is, for all $\mathbf{b}_{-i}$ and b_i, $u_i(t_i, M(t_i, \mathbf{b}_{-i})) \geq u_i(t_i, M(b_i, \mathbf{b}_{-i}))$.

Three-way and Two-way Greedy Mechanisms. We here introduce shapes of implementation trees of mechanisms for single-parameter agents. These notions have been introduced in [14, 17] and shown to either be sufficient for or characterise OSP mechanisms.

A *three-way greedy* [17] implementation allows the mechanism to interact with each agent i in one of the following ways:

- *greedy*, i.e., at each interaction the mechanism queries the agent about the best type (e.g., lowest cost in the current domain) that has not been still queried, and in case of positive answer, it assigns an outcome to the agent guaranteeing that the outcomes assigned to better types are not worse than the ones assigned to worse types;
- *reverse greedy*, i.e., at each interaction it queries the agent about the worst type (e.g., higher cost) that has not been still queried, and in case of positive answer, it assigns an outcome to the agent guaranteeing that the outcomes assigned to worse types are not better than the ones assigned to better types;
- *split & greedy*, i.e., at the first interaction it splits the agent's domain in good types (above a threshold) and bad types (below the threshold) with the guarantee that the outcome assigned in the first case is not worse than the outcome assigned in the second case; after that, the mechanism proceeds in a reverse greedy way for the good types if the agent declared to have these types, and in a greedy way among bad types otherwise.

Note that three-way greedy implementations allow mechanisms to interact with different agents in a different way.

Theorem 2.1 ([17]). *For every single-parameter setting, if a mechanism M has a three-way greedy implementation, then there are payments such that M is OSP.*

A *two-way* greedy implementation [14] is essentially a three-way greedy implementation that only allows greedy and reverse greedy interactions. However, the following variant of two-way greedy implementations turns out to have special properties for binary allocation problems.

Definition 2.2 (Two-way greedy Implementation with Interleaving). In a two-way greedy implementation with interleaving, the mechanism that interacts greedily with agent i at node u interleaves only when in $D_i^{(u)}$ all types t except the smallest one are such that $f_i(t, \mathbf{b}_{-i}) = 0$ for all $\mathbf{b}_{-i} \in D_{-i}^{(u)}$, while for the smallest type t^* there are $\mathbf{b}_{-i}, \mathbf{b}'_{-i} \in D_{-i}^{(u)}$ such that $f_i(t^*, \mathbf{b}_{-i}) = 0$ and $f_i(t, \mathbf{b}'_{-i}) = 1$. In this case, the mechanism asks all types in $D_i^{(u)} \setminus \{t^*\}$ in a reverse greedy way before making any other query to any agent.

Similarly, in a two-way greedy implementation with interleaving, the mechanism that interacts reverse greedily with agent i at node u interleaves only when in $D_i^{(u)}$ all types t except the largest one are such that $f_i(t, \mathbf{b}_{-i}) = 1$ for all $\mathbf{b}_{-i} \in D_{-i}^{(u)}$, while for the largest type t^* there are $\mathbf{b}_{-i}, \mathbf{b}'_{-i} \in D_{-i}^{(u)}$ such that $f_i(t^*, \mathbf{b}_{-i}) = 0$ and $f_i(t, \mathbf{b}'_{-i}) = 1$. In this case, the mechanism asks all types in $D_i^{(u)} \setminus \{t^*\}$ in a greedy way before making any other query to any agent.

In words, a two-way greedy Implementation with interleaving can go from a greedy to a reverse greedy phase (or viceversa) for bidder i only when the outcome for i is essentially determined (i.e., the threshold separating winning bids from losing bids has been determined at that particular history).

Theorem 2.3 ([14]). *For every single-parameter setting with binary outcomes, if a mechanism M has a two-way greedy implementation with interleaving, then there are payments such that M is OSP. Moreover, for every OSP mechanism $M = (f, p, \mathcal{T})$ there is a mechanism $M' = (f, p, \mathcal{T}')$ that has a two-way greedy implementation with interleaving.*

Both Theorems 2.1 and 2.3 are essentially existential, since they do not provide explicit payments which guarantee that the mechanisms are OSP. The existence of the payments follows from [13], where it has been proved that OSP can be characterized in terms of absence of negative-weight cycles in a suitably defined weighted graph over the possible strategy profiles. The payment for a particular player are defined therein as the shortest path in this 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. In this paper we instead show that payments in three-way greedy mechanisms and in two-way mechanisms with interleaving have a very simple structure, that makes OSP mechanisms easier to implement in practice and be understood more explicitly.

Individual Rationality and Payments. One important property we want is the following.

Definition 2.4. A mechanism M is said to satisfy individual rationality (IR) if the utility of a truthtelling agent is not negative. That is, for each agent i with type t_i, $u_i(t_i, M(t_i, \mathbf{b}_{-i})) \geq 0$ for each $\mathbf{b}_{-i}$.

An individually rational mechanism will then pay a non-negative (non-positive, resp.) amount agents whose costs are non-negative (non-positive, resp.).

There are also properties we want from payments, that depend on the sign of the agents' cost functions. Let us start with an auction setting wherein costs are negative, i.e., agents have valuations.

Definition 2.5. A mechanism M is said to be budget balanced (BB) if the sum of the payments is always non-positive. That is, for each bid profile $\mathbf{b}$, $\sum_{i \in N} p_i(\mathbf{b}) \leq 0$.

Recall that payments are added to the utility function of an agent. Therefore, the designer of a budget balanced mechanism does not have to subsidize the market, a property which is desirable in the context of mechanisms allocating items to bidders.

The second property is more appropriate to procurement auctions, that is, agents pay a non-negative cost to implement the solution computed by the mechanism.

Definition 2.6. A mechanism M is said to be normalized if for each bid profile $\mathbf{b}$ such that $f_i(\mathbf{b}) = 0$, $p_i(\mathbf{b}) = 0$.

As in the case of strategyproofness, payments can be shifted by a constant without destroying the OSP incentive compatibility. (Technically speaking, for SP, shifts for player i can depend on the bid profile of the other agents, whereas for OSP the shift can depend on the particular implementation tree adopted.) In particular, we can always add or subtract a constant high enough for our finite domains and guarantee IR. However, this shift may destroy either BB for valuations or normalization for costs. While it is known that there are shifts that make a SP mechanism both IR and BB/normalized [2], we will show that for certain OSP implementations we will need to sacrifice either IR or BB/normalization. A similar dichotomy was observed already in [18] for strategyproof combinatorial auctions that use some form of verification of bidders' behaviour [24].

3 THREE-WAY GREEDY MECHANISMS

In strategyproof mechanisms, the payments depend only on the outcome received by agent i (and thus only on the equivalence class to which the resulting bid profile belongs) and $\mathbf{b}_{-i}$. We will next show that this is not the case for OSP mechanism, since different payments can be assigned to the same outcome if they are returned at different levels of the implementation tree.

To this aim, let M be a mechanism with a three-way greedy implementation for a social function f. In order to define payments for this mechanism to be OSP, let us first introduce some useful concepts. Namely, 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 $X >_i Y$, 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). We have the following proposition.

Proposition 3.1. *Let M be a mechanism with a three-way greedy implementation and let i be an agent interacting with M in a reverse greedy way. Then truthfulness is an obvious dominant strategy for i if for every $\mathbf{b} \in X_i^k$*

$$p_i(\mathbf{b}) = \theta_{\mathbf{b}}(k) f_i^k + \sum_{j=0}^{k-1} (\theta_{\mathbf{b}}(j) - \theta_{\mathbf{b}}(j+1)) f_i^j.$$

Proof. Let t be the type of i, and let q be the type which i is queried about. Let f_i^o and p_i^o be respectively the outcome and the payment that would be assigned by the mechanism to i if she positively answers to the query.

We now distinguish multiple cases. If $q > t$, then the truthful action would be to negatively answer to the query. Suppose that the worst possible utility received by taking this truthful action occurs when agent i receives outcome f_i^k with $k \geq o$, and payment p_i^k. Note that $\theta_{(t, \mathbf{b}_{-i})}(k)$ and $\theta_{(q, \mathbf{b}'_{-i})}(k) = \theta(k)$ for every k, and every $\mathbf{b}_{-i}$ and $\mathbf{b}'_{-i}$ compatible with the currently considered query, since the profiles $(t, \mathbf{b}_{-i})$ and $(q, \mathbf{b}'_{-i})$ are related because separated by i. Thus the difference between the utility received in answering truthfully and in answering untruthfully would be $\Delta u = p_i^k - t f_i^k - p_i^o + t f_i^o$. Clearly, if $f_i^o = f_i^k$, then also $p_i^k = p_i^o$, and hence $\Delta u = 0$. Suppose instead that $o < k$, and thus $f_i^o < f_i^k$. Then

$$\begin{aligned}
\Delta u &= \theta(k) f_i^k + \sum_{j=0}^{k-1} (\theta(j) - \theta(j+1)) f_i^j - t f_i^k \\
&\quad - \theta(o) f_i^o - \sum_{j=0}^{o-1} (\theta(j) - \theta(j+1)) f_i^j + t f_i^o \\
&= f_i^k (\theta(k) - t) - f_i^o (\theta(o) - t) + \sum_{j=o}^{k-1} (\theta(j) - \theta(j+1)) f_i^j
\end{aligned}$$

$$\geq f_i^k(\theta(k)-t) - f_i^o\left[(\theta(o)-t) + \sum_{j=o}^{k-1}(\theta(j+1)-\theta(j))\right]$$
$$= (f_i^k - f_i^o)(\theta(k)-t) \geq 0.$$

The case for $q = t$ is symmetric. This concludes the proof since, by observations above, it never occurs that $q < t$. □

It is immediate to check that payments defined in Proposition 3.1 are essentially the same as strategyproof payments as defined in [2]. Hence, we achieve that payments for agents interacting in a reverse greedy way can be chosen in a way so that these agents are truthful and individual rational. Moreover, if all agents would be queried in this way the mechanism is clearly budget-balanced.

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 in Proposition 3.1, but they fail to match SP payments.

Proposition 3.2. *Let M be a mechanism with a three-way greedy implementation tree and let i be an agent interacting with 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 $\mathbf{b} \in X_i^k$*

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

Proof. Let t be the type of i, and let q be the type which i is queried about. Let f_i^o and p_i^o be respectively the outcome and the payment that would be assigned by the mechanism to i if she positively answers to the query. We now distinguish multiple cases. If $q < t$, then the truthful action would be to negatively answer to the query. Suppose that the worst possible utility received by taking this truthful action occurs when agent i receives outcome f_i^k with $k \leq o$, and payment p_i^k. Thus the difference between the utility received in answering truthfully and in answering untruthfully would be $\Delta u = p_i^k - t f_i^k - p_i^o + t f_i^o$. We further distinguish two subcases. Consider first that $f_i^k > f_i^0$. Clearly, if $f_i^o = f_i^k$, then also $p_i^k = p_i^o$, and hence $\Delta u = 0$. Suppose instead that $k < o$, and thus $f_i^o > f_i^k$. Then,

$$\Delta u = \tau(k) f_i^k + \sum_{j=0}^{k-1} (\tau(j) - \tau(j+1)) f_i^j - t f_i^k$$
$$- \tau(o) f_i^o - \sum_{j=0}^{o-1} (\tau(j) - \tau(j+1)) f_i^j + t f_i^o$$
$$= f_i^k(\tau(k)-t) - f_i^o(\tau(o)-t) - \sum_{j=k}^{o-1} (\tau(j) - \tau(j+1)) f_i^j$$
$$\geq f_i^k(\tau(k)-t) - f_i^o\left[(\tau(o)-t) + \sum_{j=k}^{o-1}(\tau(j)-\tau(j+1))\right]$$
$$= (f_i^k - f_i^o)(\tau(k)-t) \geq 0.$$

Suppose now that $f_i^k = f_i^0$. If also $f_i^o = f_i^0$, then i cannot receive outcome larger than f_i^0 both answering negatively and positively to the query. Hence, any $b'_i \leq t$ for which there is $\mathbf{b}'_{-i}$ such that $\mathbf{b}' \in X_i^{>0}$ is also smaller than q, and hence $p_i^k = p_i^o$, from which it follows that $\Delta u = 0$.

Suppose instead that $f_i^o > 0$. Then q is smaller than t and there is $\mathbf{b}'_{-i}$ such that $(q, \mathbf{b}'_{-i}) \in X_i^{>0}$, and thus p_i^k must be at most $p_i^o - t f_i^o$, from which it is immediate that $\Delta u \geq 0$.

If $q = t$, then the truthful action would be to positively answer to the query. Suppose that the best possible utility achieved if agent i instead answers negatively occurs when this agent is receives outcome f_i^k with $k \leq o$, and payment p_i^k. Thus the difference between the utility received in answering truthfully and in answering untruthfully would be $\Delta u = p_i^o - t f_i^o - p_i^k + t f_i^k$. We further distinguish two subcases. Consider first that $f_i^k > f_i^0$. Clearly, if $f_i^o = f_i^k$, then also $p_i^k = p_i^o$, and hence $\Delta u = 0$. Suppose instead that $k < o$, and thus $f_i^o > f_i^k$. Then,

$$\Delta u = \tau(o) f_i^o + \sum_{j=0}^{o-1} (\tau(j) - \tau(j+1)) f_i^j - t f_i^o$$
$$- \tau(k) f_i^k - \sum_{j=0}^{k-1} (\tau(j) - \tau(j+1)) f_i^j + t f_i^k$$
$$= -f_i^k(\tau(k)-t) + f_i^o(\tau(o)-t) + \sum_{j=k}^{o-1} (\tau(j) - \tau(j+1)) f_i^j$$
$$\geq -f_i^k(\tau(k)-t) + f_i^o\left[(\tau(o)-t) + \sum_{j=k}^{o-1}(\tau(j)-\tau(j+1))\right]$$
$$= (f_i^o - f_i^k)(\tau(k)-t) \geq 0,$$

where the last inequality follows since $\tau(k) \geq t$. Indeed, $q(k) > t$, otherwise (i.e., if $t > q(k)$) it was not possible to assign outcome $f_i^o > f_i^k$ at agent i when queried for t, and thus , since $\tau(k) > q(k)$, the desired result follows.

Suppose now that $f_i^k = f_i^0$. If also $f_i^o = f_i^0$, then i cannot receive outcome larger than f_i^0 both answering negatively and positively to the query. Hence, any $b'_i \leq t$ for which there is $\mathbf{b}'_{-i}$ such that $\mathbf{b}' \in X_i^{>0}$ is also smaller than any b_i still available at i, and hence $p_i^k = p_i^o$, from which it follows that $\Delta u = 0$.

Suppose instead that $f_i^o > 0$. Then t is smaller than any available bid b_i and there is $\mathbf{b}'_{-i}$ such that $(t, \mathbf{b}'_{-i}) \in X_i^{>0}$, and thus p_i^k must be at most $p_i^o - t f_i^o$, from which it is immediate that $\Delta u \geq 0$.

This concludes the proof since it never occurs that $q > t$. □

Note that there are two main differences between payments as defined in Proposition 3.2 and payments provided in Proposition 3.1: 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. These differences cause the payments for agents greedily interacting with the mechanism to be different from SP payments. Specifically, in Section 5, we will provide examples in which the mechanism cannot be both individual rational and budget balanced.

We finally conclude this section by describing payments for agents interacting in split & greedy way. Recall that this consists in first asking to separate the domain in good and bad types (assuring outcomes for good types to be 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 payments also are a composition of above described payments.

Proposition 3.3. *Let M be a mechanism with a three-way greedy implementation tree and let i be an agent interacting with M in a split & greedy way. Let $X_i^0, \dots, X_i^m$ be the partition of type profiles in equivalence class for i as defined above. Let t^* be the threshold that the mechanism uses to distinguish among good and bad types, and f_i^o and f_i^x be respectively the worst outcome achievable when the agent's declared type is good and the best outcome achievable when the agent's declared type is bad (note that f_i^x is not necessarily different from f_i^o). Moreover, let $\theta(j)$ and $\tau(j)$ as defined above. Then truthfulness is an obvious dominant strategy for i if for every $\mathbf{b} \in X_i^k$ with $b_i > t^*$ (bad types)*

$$p_i(\mathbf{b}) = \begin{cases} \min\left\{0, \min_{\substack{t < b'_i \leq b_i \\ \exists \mathbf{b}'_{-i}:\, \mathbf{b}' \in X_i^{>0}}} \left(p_i(\mathbf{b}') - b'_i f_i(\mathbf{b}')\right)\right\} & \text{if } k = 0; \\ \tau(k) f_i^k + \sum_{j=0}^{k-1} (\tau(j) - \tau(j+1)) f_i^j & \text{o.w.;} \end{cases}$$

and for every $\mathbf{b} \in X_i^k$ with $b_i \leq t^*$ (good types)

$$\pi_i(\mathbf{b}) = p_i^* + \theta_{\mathbf{b}}(k) f_i^k + \sum_{j=o}^{k-1} (\theta_{\mathbf{b}}(j) - \theta_{\mathbf{b}}(j+1)) f_i^j,$$

where p_i^* is such that $\min_{\mathbf{b}}\{\pi_i(\mathbf{b})\} - \max_{\mathbf{b}}\{p_i(\mathbf{b})\} = t^*(f_i^o - f_i^x)$.

Proof. First observe that for any query except for the first one, payments are exactly the same as described above with outcomes restricted to be in $\{f_i^0, \dots, f_i^x\}$ for bad types, and in $\{f_i^o, \dots, f_i^m\}$ for good types. Hence, for each of these queries it is convenient to be truthful.

Consider then the first query (i.e., the split query). Let us start by considering agent i with type $t \leq t^*$ (good). We first prove that the best outcome that this agent can achieve if she declares to have a bad type would be f_i^x, that is the best possible outcome for bad types. Indeed, the utility achieved when the outcome is f_i^k, for $0 < k < x$, is $(\tau(k) - t) f_i^k + \sum_{j=0}^{k-1} (\tau(j) - \tau(j+1)) f_i^j$, whereas the utility achieved when the outcome is f_i^x is $(\tau(x) - t) f_i^x + \sum_{j=0}^{k-1} (\tau(j) - \tau(j+1)) f_i^j + \sum_{j=k}^{x-1} (\tau(j) - \tau(j+1)) f_i^j$. Observe that $t f_i^x = t f_i^k + t \cdot \sum_{j=k}^{x-1} (f_i^{j+1} - f_i^j)$ and $\tau(x) f_i^x + \sum_{j=k}^{x-1} (\tau(j) - \tau(j+1)) f_i^j = \tau(k) f_i^k + \sum_{j=k}^{x-1} \tau(j+1)(f_i^{j+1} - f_i^j)$. Hence, we have that the utility achieved when the outcome is f_i^x is

$$(\tau(k) - t) f_i^k + \sum_{j=0}^{k-1} (\tau(j) - \tau(j+1)) f_i^j + \sum_{j=k}^{x-1} (\tau(j+1) - t)(f_i^{j+1} - f_i^j),$$

that is clearly larger than the one for outcome f_i^k, for $0 < k < x$, and it is larger than 0, and thus larger than the utility that would be achieved for outcome f_i^0.

Now observe that, by OSPness of reverse greedy algorithms, the worst truthful utility for i if she plays truthfully is at least as good as the best utility she would achieve if she behaves as if her type was good but untruthful, that in turns is at least as good as receiving the smallest possible payment assigned during the reverse greedy phase and receiving the corresponding outcome. Let $U = \min_{\mathbf{b}}\{\pi_i(\mathbf{b})\} - t f_i^o$ be this latter value. The utility that is received by i when she received outcome f_i^x and the corresponding payment is instead $\max_{\mathbf{b}}\{p_i(\mathbf{b})\} - t f_i^x = \min_{\mathbf{b}}\{\pi_i(\mathbf{b})\} - t^*(f_i^o - f_i^x) - t f_i^x = U + (t - t^*)(f_i^o - f_i^x) \leq U$. Since, as showed above, this outcome is the one that maximizes the utility of i if she untruthfully answers to the split query, we have that i has no incentive to give this untruthful answer.

Suppose now that agent i has type $t > t^*$ (bad). We first prove that the best outcome that this agent can achieve if she declares to have a good type would be f_i^o, that is the worst possible outcome for these types. Indeed, let $\check{\theta}(k) = \min_{\mathbf{b} \in X_i^k} \theta_{\mathbf{b}}(k)$, for every $k \geq o$. Then, the utility achieved when the outcome is f_i^o is $p_i^* - (t - t^*) f_i^o$, whereas the utility achieved when the outcome is f_i^k, for $k > o$ is $p_i^* - (t - \check{\theta}(k)) f_i^k + \sum_{j=o}^{k-1} (\check{\theta}(j) - \check{\theta}(j+1)) f_i^j$. Observe that $t f_i^k = t f_i^o + t \cdot \sum_{j=o}^{k-1} (f_i^{j+1} - f_i^j)$ and $\check{\theta}(k) f_i^k + \sum_{j=o}^{k-1} (\check{\theta}(j) - \check{\theta}(j+1)) f_i^j = \check{\theta}(o) f_i^o \sum_{j=o}^{k-1} \check{\theta}(j+1)(f_i^{j+1} - f_i^j)$. Hence, we have that the utility achieved when the outcome is f_i^k is, as desired,

$$p_i^* - (t - \check{\theta}(o)) f_i^o - \sum_{j=o}^{k-1} (t - \check{\theta}(j+1))(f_i^{j+1} - f_i^j) \leq p_i^* - (t - t*) f_i^o.$$

Now observe that, by OSPness of greedy algorithms, the worst truthful utility for i if she plays truthfully is at least as good as the best utility she would achieve if she behaves as if her type was bad but untruthful, that in turns is at least as good as receiving the largest possible payment assigned during the greedy phase and receiving the corresponding outcome. Let $U = \max_{\mathbf{b}}\{p_i(\mathbf{b})\} - t f_i^x$ be this latter value. The utility that is received by i if she receives the lowest outcome for good types and the corresponding payment is instead $\min_{\mathbf{b}}\{\pi_i(\mathbf{b})\} - t f_i^o = \max_{\mathbf{b}}\{p_i(\mathbf{b})\} + t^*(f_i^o - f_i^k) - t f_i^o = U + (t^* - t)(f_i^o - f_i^k) \leq U$. Since, as showed above, this is the outcome that maximizes the utility of i if she untruthfully answers the split query, then i has no incentive to give this untruthful answer. □

4 BINARY OUTCOMES OSP MECHANISMS

Let us consider here setting with binary outcomes. In this case it is possible to largely simplify the payment rules described above. Specifically, for reverse greedy, we have the following rule:

pay 0 to losers and $\theta(\mathbf{b})$ in every profile $\mathbf{b}$ in which i wins. (1)

As for greedy, it is useful to give some preliminary definitions. Given two winning types u and u' asked to i, we say that there is a *winning phase* between u and u' if for all queries (to agent i or other agents) $\hat{u}^1, \ldots, \hat{u}^\ell$ between the queries for u and u', i receives the winning outcome in every leaf of the subtrees rooted in the child of $\hat{u}^j$ different from $\hat{u}^{j+1}$ for every $j = 1, \ldots, \ell$ with $\hat{u}^{\ell+1} = u'$. Then, let u_0 be the first winning type asked to i. If there is a winning phase starting in u_0, then let u'_0 be the other extreme of this phase, otherwise let $u'_0 = u_0$. Similarly, for $j > 0$, let u_j be the first winning type asked to i after u'_{j-1}, and if there is a winning phase starting in u_j, let u'_j be the other extreme of this phase, otherwise let $u'_j = u_j$. We then let P_1 be the sequence of queries from the root of $\mathcal{T}$ to the node representing the u_1 query, and P_j be the the sequence of queries from u'_{j-1} to u_j. Hence we can partition profiles $\mathbf{b}$ with a losing outcomes for i as follows: we say that a losing profile $\mathbf{b}$ is in L_j if $\mathbf{b}$ corresponds to a leaf of the subtree rooted in a node $\hat{u} \notin P_j$ which is the child of some $u \in P_j$. See Figure 1 for an illustration of these concepts.

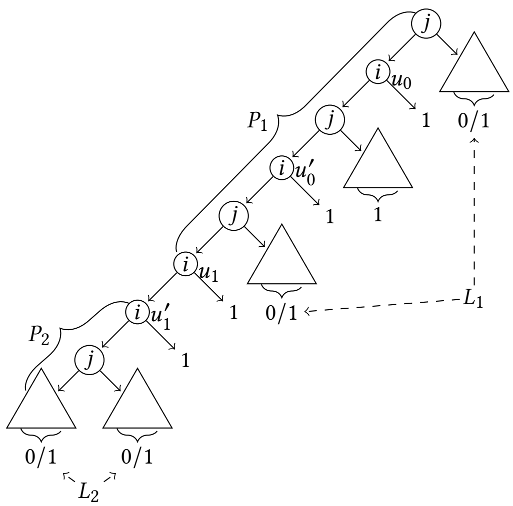

Figure 1: An example of implementation tree. Here the label of a node defines the agent that is queried at that time step; the label on edges are the type that agent i are queried about; triangles are used for representing arbitrary subtrees in which no query has been done to i; label 1 means that agent i will surely receive outcome 1 in the corresponding subtree, and label $0/1$ means that she can receive both outcome 0 or 1.

The payment received by agent i that is queried in a greedy way, is then as follows:

$$\begin{cases} \tau(1), & \text{if she wins,} \\ 0, & \text{if she loses in a profile } \mathbf{b} \in L_1, \\ \tau(1) - u'_j, & \text{if she loses in a profile } \mathbf{b} \in L_j. \end{cases} \tag{2}$$

We next show that these payments not only work for mechanisms with three-way (and thus also two-way) greedy implementation, but also for two-way greedy implementation with interleaving.

Proposition 4.1. *Let M be a mechanism for binary outcomes with a two-way greedy with interleaving implementation tree and let i be an agent interacting with M. Then with payments as in Proposition 3.1 and Proposition 3.2, truthfulness is obvious dominant for i.*

Proof. Consider first the case that i is queried in a reverse greedy way. If interleaving does not occur (i.e. either interleaving is not prescribed or the true type t of i is larger than the threshold at which interleaving would occur), then the claim follows from Proposition 3.1. Otherwise, let x be the maximum available type of i when interleaving occurs. Note that for every type $q \leq x$ agent i may achieve winning outcome and for every $q < x$ agent i cannot receive a winning outcome. Hence, for every $\mathbf{b}$ that is available when interleaving occurs we have that $\theta(\mathbf{b}) = x$. Note also that the agent is only queried about types $q < x$, since as soon as all these queried has been done, then the type of i is revealed to be x. Thus, agent i of type t when queried about type $q < x$ after interleaving receives utility $x - t$ if she answers positively. Moreover, a negative answer will guarantee utility $x - t$ if $t < x$, and utility 0 otherwise. Thus, agent i is indifferent from answering positively or negatively to queries: indeed, for $t < x$ both the worst truthful and the best untruthful outcome are winning outcomes, and thus they guarantee the same utility $x - t$; for $t = x$ instead the worst truthful outcome may be losing, but still the utility for losing is in this case the same as the utility for winning.

Consider now the case that i is queried in a greedy way. If interleaving does not occur (i.e. either interleaving is not prescribed or the true type t of i is smaller than the threshold at which interleaving would occur), then the claim follows from Proposition 3.1. Otherwise, let x be the minimum available type of i when interleaving occurs. First observe that, according to the definition of payments given in Proposition 3.1, the payment for losing outcomes in the last subset L_i of losing outcomes must be $\tau(1) - x$, since x is among the values for which the minimum can be computed, and this minimum is clearly achieved by x since for every $q > x$, the term to minimize would be $\tau(1) - q > \tau(1) - x$. Finally, note that, since after interleaving there is no query assigning winning outcome, any profile $\mathbf{b}$ for which i receives a losing outcome after interleaving belongs to L_i, and thus it receives payment $\tau(1) - x$.

Note that for every type $q \geq x$ agent i may achieve a losing outcome. Hence, $\tau(1) \leq x$. Thus, agent i of type t when queried about type $q > x$ after interleaving receives utility $\tau(1) - x$ if she answers positively. However, a negative answer will guarantee the same utility: this is immediately follows from previous observations whenever the outcome is achieved in a profile $\mathbf{b}$ with $b_i > x$, that must necessarily be a profile corresponding to a losing outcome; if $b_i = x$, then it is possible both to have a profile corresponding to a losing outcome, still guaranteeing utility $\tau(1) - x$, and one corresponding to a winning outcome, that still provides the same utility. Thus, whatever the value of t is, truthful and untruthful actions will guarantee the same utility. □

Then, from Theorem 2.3, we achieve the following corollary.

Corollary 4.2. *A mechanism $M = (f, p, \mathcal{T})$ is OSP if and only if payments are as defined in* (1) *and* (2)*, and the implementation tree $\mathcal{T}$ is (or can be transformed to) two-way greedy with interleaving.*

We next provide a very simple example that highlights the peculiarities of the greedy case. Consider a single-item auction with two bidders and valuations for the item in the set $\{H, M, L, B\}$ with $H > M > L > B$. Here, a greedy mechanism M will first ask bidder 1 if her valuation is H, and, in case of positive answer, assign the item to her. Then M asks bidder 2 whether her valuation is H, and, in case of positive answer, assign the item to bidder 2. M proceeds similarly for valuations M, L and B in the domain. (Note that with only two possible valuations these last two queries are unneeded, since we already know that the valuation is B for both bidders and we can sell the item to either. However, we here make those queries explicit for sake of clarity and completeness.) Looking at Figure 1, we have that the u nodes are equal to the u' nodes for both bidders; at u_0 they separate H from $\{M, L, B\}$; at u_1, they split M from $\{B, L\}$ whilst at u_2 they separate B from L.

From our definition, either bidder will be charged M when she wins the item. Bidder 1 will be charged 0 when she loses. Bidder 2 instead will be charged 0 when she loses and the winner signalled type H, M or L, and $M - L$ otherwise. The intuition for these payments is the following; let us focus on bidder 2 w.l.o.g.. The payment p for winning in profiles $(\cdot, H)$ must be at least M. Let us focus on the first divergence of bidder 2 at u_0 and assume that her true valuation is M. When she is honest (and takes the M, L, B branch) her worst utility is at most 0 (this is when she loses to bidder 1 with valuation M) whereas her best utility for lying (i.e., taking the H branch) would be $M - p$. Then for the mechanism to be OSP, we need that $M - p$ is at most 0. The payment of bidder 2 for winning cannot be lower than M also for other profiles. Assume not and consider she has valuation H; she would lie at u_0 since her best lie would guarantee better utility than being honest. Finally, we cannot charge bidder 2 nothing for the profile (B, B) where she loses due to the following argument: assume she has valuation L, and thus her utility is $L - M < 0$ when she wins in (L, B). If we did not charge her for (B, B) then at u_2, bidder 2 would be better off lying and take the B branch.

5 SINGLE-MINDED AUCTIONS

In this section we apply the results of the previous section to the well-known setting of combinatorial auctions with single-minded bidders. Here, we are given a set U of m items and a set N of n agents. Each agent i has a private *valuation* function v_i and is interested in obtaining only one particular subset of U; we denote i's desired bundle of item by R_i. We assume that R_i is public knowledge.

The objective is social welfare maximization, that is, finding a partition $(S_1, \dots, S_n)$ of U such that $\sum_{i=1}^{n} v_i(S_i)$ is maximized. This is a problem, which is NP-hard to approximate better than $\sqrt{m}$, bound that can be guaranteed via a greedy algorithm, due to [19]. The algorithm simply sorts bidders by efficiency (defined as valuation for the desired bundle over the square root of the bundle size) and grants sets to bidders in this order as long as feasibility is guaranteed. [9] shown that there is an OSP implementation for this (Algorithm 1).

Recall that the payments we defined above for greedy and binary outcomes would charge bidders even if they end up losing their wanted set R_i. Moreover, the payments above could even charge a winning bidder more than their valuation for the set they obtain. In order to enforce voluntary participation (that is, honest bidders will not incur into loss by participating to the auction) we envision the auctioneer subsidising these bidders and rescale all the payments in a way that no losing bidder ends up paying for participating. Next we bound the subsidies needed to implement Algorithm 1.

Algorithm 1: Greedy implementation of algorithm in [19]

Define function Φ_i as $\Phi_i(x) = x/\sqrt{|R_i|}$.
$\mathcal{P} \leftarrow \emptyset$ (the set of bundles that have already been allocated)
$\mathcal{N} \leftarrow N$ (the set of agents currently under consideration)
$\mathcal{D}_i \leftarrow D_i \ \forall i \in N$ (values in i's domain under consideration)
while $\mathcal{N} \neq \emptyset$ **do**
  Let $j = \arg\max_{k \in \mathcal{N}} \Phi_k(\max \mathcal{D}_k)$
  if *there is S in $\mathcal{P}$ s.t. $R_j \cap S \neq \emptyset$* **then** $\mathcal{N} = \mathcal{N} \setminus \{j\}$;
  else
    Ask j if her valuation is $\max \mathcal{D}_j$
    if yes **then** $\mathcal{P} \leftarrow \mathcal{P} \cup \{R_j\}$ and $\mathcal{N} = \mathcal{N} \setminus \{j\}$;
    else $\mathcal{D}_j = \mathcal{D}_j \setminus \{\max \mathcal{D}_j\}$;
Return $\mathcal{P}$

Let us for simplicity consider the case in which $D_i = D = \{v_1, \dots, v_\ell\}$ with $v_1 > v_2 > \dots > v_\ell$. We have the following bound on the subsidies needed by the greedy algorithm for single-minded combinatorial auctions.

Theorem 5.1. *Algorithm 1 needs subsidies of at most $n \cdot v_2 - (n-1) \cdot v_\ell - v_{\ell-1}$.*

Proof. Consider the setting in which $m = n - 1$ with $U = \{x_1, \dots x_m\}$, and $R_i = \{x_i\}$ for all $i = 1, \dots, n-1$ whereas $R_n = U$. Furthermore, assume that D is such that $v_j > \frac{v_j}{\sqrt{m}} > v_{j+1}$ for all $j \in \{1, \dots, \ell - 1\}$. In other words, this setting is akin a knapsack auction with only two feasible solutions, one comprised of the first $n - 1$ bidders and the other containing bidder n only. The implementation tree built by Algorithm 1 will query all bidders (w.l.o.g. in order of their ids) for v_1 first, followed by a sequence of queries for v_2 and so on up to queries for $v_{\ell-1}$. We observe that for this implementation tree, there are winning phases comprising a single node, i.e., $u_j = u'_j$ for all $j = 0, \dots, \ell - 1$. Moreover, note that $\tau(1) = v_2$ meaning that a winning bidder would be charged v_2 for the bundle, whereas a losing bidder i would be charged $\tau(1) - v_k$ if her last query was for v_k. Consider, therefore, the execution of the mechanism for bid profile $\mathbf{b}$ in which $b_i = v_\ell$ for all i. Here the winning bidders will be charged v_2, making their utility equal to $v_\ell - v_2 < 0$. The losing bidder will instead be charged $v_2 - v_{\ell-1}$. This proves that a subsidy of $(n-1)(v_2 - v_\ell) + v_2 - v_{\ell-1}$ is needed.

To see that there is no instance wherein the auctioneer could need larger subsidies we make two easy observations. Firstly, $\tau(1)$ is always upper bounded by v_2. Secondly, losing bidders cannot be charged more than $\tau(1) - v_k$, v_k being the last query they received, and so the charge cannot be larger than $v_2 - v_{\ell-1}$. □

Theorem 5.2. *There exist instances for which Algorithm 1 needs no subsidies.*

Proof Sketch. Suppose that $\frac{v_\ell}{\sqrt{|R_i|}} > \frac{v_1}{\sqrt{|R_{i+1}|}}$ for $i = 1, \dots, n-1$. In this case, bidder 1 would first be queried for all the valuations in the domain, followed by the same series of queries to bidder 2 and so on so forth. Then, the payment for winning will be $\tau(1) = v_\ell$ and the payment for losing will be 0. □

ACKNOWLEDGMENTS

Diodato Ferraioli was partially supported by the Italian MIUR PRIN2017 Project ALGADIMAR "Algorithms, Games, and Digital Markets" and by "GNCS–INdAM". Carmine Ventre acknowledges funding from the UKRI Trustworthy Autonomous Systems Hub (EP/V00784X/1).

REFERENCES

[1] Marek Adamczyk, Allan Borodin, Diodato Ferraioli, Bart De Keijzer, and Stefano Leonardi. 2017. Sequential posted-price mechanisms with correlated valuations. *ACM Transactions on Economics and Computation (TEAC)* 5, 4 (2017), 1–39.

[2] Aaron Archer and Éva Tardos. 2001. Truthful Mechanisms for One-Parameter Agents. In *FOCS '01*. 482–491.

[3] R Pablo Arribillaga, Jordi Massó, and Alejandro Neme. 2019. All sequential allotment rules are obviously strategy-proof. (2019).

[4] R Pablo Arribillaga, Jordi Massó, and Alejandro Neme. 2020. On obvious strategy-proofness and single-peakedness. *Journal of Economic Theory* 186 (2020), 104992.

[5] Itai Ashlagi and Yannai A. Gonczarowski. 2018. Stable matching mechanisms are not obviously strategy-proof. *Journal of Economic Theory* 177 (2018), 405–425.

[6] Moshe Babaioff, Nicole Immorlica, Brendan Lucier, and S Matthew Weinberg. 2020. A simple and approximately optimal mechanism for an additive buyer. *Journal of the ACM (JACM)* 67, 4 (2020), 1–40.

[7] Sophie Bade and Yannai A. Gonczarowski. 2017. Gibbard-Satterthwaite Success Stories and Obvious Strategyproofness. In *EC '17*. 565.

[8] Giorgos Christodoulou, Vasilis Gkatzelis, and Daniel Schoepflin. 2022. Optimal Deterministic Clock Auctions and Beyond. In *ITCS '22*, Mark Braverman (Ed.), Vol. 215. 49:1–49:23.

[9] Bart de Keijzer, Maria Kyropoulou, and Carmine Ventre. 2020. Obviously Strategyproof Single-Minded Combinatorial Auctions. In *ICALP '20*. 71:1–71:17.

[10] Paul Dütting, Vasilis Gkatzelis, and Tim Roughgarden. 2017. The Performance of Deferred-Acceptance Auctions. *Math. Oper. Res.* 42, 4 (2017), 897–914.

[11] Michal Feldman, Vasilis Gkatzelis, Nick Gravin, and Daniel Schoepflin. 2022. Bayesian and Randomized Clock Auctions. In *EC '22*. 820–845.

[12] Diodato Ferraioli, Adrian Meier, Paolo Penna, and Carmine Ventre. 2019. Automated Optimal OSP Mechanisms for Set Systems - The Case of Small Domains. In *WINE '19*. 171–185.

[13] Diodato Ferraioli, Adrian Meier, Paolo Penna, and Carmine Ventre. 2019. Obviously Strategyproof Mechanisms for Machine Scheduling. In *ESA '19*. 46:1–46:15.

[14] Diodato Ferraioli, Paolo Penna, and Carmine Ventre. 2021. Two-Way Greedy: Algorithms for Imperfect Rationality. In *WINE '21*. 3–21.

[15] Diodato Ferraioli and Carmine Ventre. 2017. Obvious strategyproofness needs monitoring for good approximations. In *AAAI '17*.

[16] Diodato Ferraioli and Carmine Ventre. 2018. Probabilistic Verification for Obviously Strategyproof Mechanisms. In *IJCAI '18*. 240–246.

[17] Diodato Ferraioli and Carmine Ventre. 2023. On the Connection between Greedy Algorithms and Imperfect Rationality. https://doi.org/10.48550/ARXIV.2302.13641

[18] Piotr Krysta and Carmine Ventre. 2015. Combinatorial auctions with verification are tractable. *Theor. Comput. Sci.* 571 (2015), 21–35.

[19] Daniel Lehmann, Liadan Ita O'Callaghan, and Yoav Shoham. 2002. Truth Revelation in Approximately Efficient Combinatorial Auctions. *J. ACM* 49, 5 (2002), 577–602.

[20] Shengwu Li. 2017. Obviously Strategy-Proof Mechanisms. *American Economic Review* 107, 11 (2017), 3257–87.

[21] Andrew Mackenzie. 2020. A revelation principle for obviously strategy-proof implementation. *Games and Economic Behavior* 124 (2020), 512–533.

[22] P. Milgrom and I. Segal. 2020. Clock Auctions and Radio Spectrum Reallocation. *Journal of Political Economy* (2020).

[23] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay Vazirani (Eds.). 2017. *Algorithmic Game Theory*.

[24] Paolo Penna and Carmine Ventre. 2014. Optimal collusion-resistant mechanisms with verification. *Games Econ. Behav.* 86 (2014), 491–509.

[25] Marek Pycia and Peter Troyan. 2019. Obvious Dominance and Random Priority. In *EC '19*.