Implementation with Contingent Contracts *

Rahul Deb † and Debasis Mishra ‡

June 7, 2014

Abstract

We study dominant strategy incentive compatibility in a mechanism design setting with contingent contracts where the payoff of each agent is observed by the principal and can be contracted upon. Our main focus is on the class of linear contracts (one of the most commonly used contingent contracts) which consist of a transfer and a flat rate of profit sharing. We characterize outcomes implementable by linear contracts and provide a foundation for them by showing that, in finite type spaces, every social choice function that can be implemented using a more general nonlinear contingent contract can also be implemented using a linear contract. We then qualitatively describe the set of implementable outcomes. We show that a general class of social welfare criteria can be implemented. This class contains social choice functions (such as the Rawlsian) which cannot be implemented using (uncontingent) transfers. Under additional conditions, we show that only social choice functions in this class are implementable.

1 Introduction

The classic setting in mechanism design with quasi-linear payoffs is the following. Agents privately observe their types and make reports to the principal. Based on these reports, the principal chooses an alternative and transfer amounts. Agents then realize their payoff from the chosen alternative and their final payoff is this payoff less their transfer amount. We refer to such mechanisms as quasilinear mechanisms. An important feature of this setting is that the mechanism is a function only of the reports and not of the realized payoffs of the

*This paper is a significant revision of an earlier paper titled "Implementation with Securities". We are grateful to the co-editor Matthew Jackson and three anonymous referees for their insightful comments and suggestions which greatly improved the paper. We are also grateful to Juan Carlos Carbajal, Dan Quint, Arunava Sen, Andy Skrzypacz, Rakesh Vohra and numerous seminar audiences for valuable feedback.

†University of Toronto. Email: rahul.deb@utoronto.ca

‡Indian Statistical Institute, Delhi. Email: dmishra@isid.ac.in

agents. This could either be because the principal cannot observe these payoffs or that they are not verifiable by third parties and hence contracts based on them cannot be enforced.

However, in many practical settings, principals use *contingent contracts*. An agent's payoff from such a contract depends not only on the reported types but also on the realized payoff (which is observed and contractible). Perhaps the simplest and most commonly observed example of a contingent contract is a *linear contract*. Here, the contract consists of a lump sum transfer and a flat percentage (such as a royalty rate or a tax) which determines how the principal and an agent share the latter's payoff. Though such contracts can be useful in settings where agents are cash constrained, they primarily serve the purpose of providing a larger set of tools using which a principal can incentivize the agents.

Indeed, contingent contracts are ubiquitous and settings where they are used include publishing agreements with authors, musicians seeking record labels, the sale of patents, entrepreneurs selling their firms to acquirers or soliciting venture capital and sports associations selling broadcasting rights. They are employed in the form of taxes and tolls to finance public goods provision. Auctions are often conducted in which buyers bid using such contracts as opposed to simply making cash bids. Examples include the sale of private companies and divisions of public companies, government sales of oil leases, wireless spectrum and highway building contracts.[1]

In this paper, our aim is to study dominant strategy implementation using contingent contracts. In our model, the agents first report their types to the principal who then chooses an alternative using a social choice function (scf) which depends on these reports. The contractible stochastic payoff of an agent, the distribution of which depends on the agent's true type and the chosen alternative, is then realized. The final payoff to an agent from the contingent contract is an increasing function of his realized payoff and the vector of types reported by all the agents. A mechanism in this context consists of an scf and a contingent contract for each agent. We say that an scf is implementable (using a linear contract) if there exists a (linear) contingent contract such that truthful reporting of type is a dominant strategy for each agent in the resulting mechanism.

Surprisingly, we show that any scf implementable using a general nonlinear contingent contract can also be implemented using a linear contract. Put differently, this result states that the set of scfs implementable by linear contracts is not expanded by using contingent contracts that depend nonlinearly on the realized payoff of the agents. This result can be interpreted as a foundation for linear contracts and provides one explanation for their ubiquity in practical applications. Further, we show that the set of scfs implementable by linear contracts is characterized by a condition called *acyclicity*, which is simple to interpret

[1]While our focus is on settings with adverse selection, it should be pointed out that contingent contracts are also typical in problems with moral hazard where an agent's final payoff depends on the observed output.

and apply.

It is natural to expect that, in an environment where the principal can contract on the realized payoffs, the set of implementable outcomes is larger than those that can be attained by simply using (uncontingent) transfers. We identify a family of scfs called *aggregate payoff maximizers* and show that they are acyclic, and hence, implementable. Examples of aggregate payoff maximizers include the efficient and the Rawlsian (or max-min) scfs, of which the latter is known to be not implementable using transfers alone. When the type space satisfies an additional richness condition, we show that the only implementable scfs satisfying an independence condition are the aggregate payoff maximizers. Thus, under these additional conditions, we provide a complete, qualitative description of the set of implementable scfs.

Despite the implementability equivalence between linear and contingent contracts, it should be pointed out that *payoff equivalence* between these contract forms does not hold (we provide an simple example demonstrating this). However, we provide a simple argument which yields an important property of the payoffs achievable by linear contracts. They can always be used to achieve efficiency with budget balance overcoming the known budget deficit shortcoming of Vickrey-Clarke-Groves transfers (Green and Laffont, 1979).

1.1 Related Literature

This paper is related to a few different strands of literature. Mechanism design with contingent contracts originated with the literature on security auctions (Hansen, 1985), a recent survey is Skrzypacz (2013). This paper has been partly inspired by the recent work which discuss the revenue ranking of auctions conducted with different contingent contracts (DeMarzo et al., 2005; Che and Kim, 2010; Abhishek et al., 2013). These papers study how a seller's revenue is affected by the "steepness" of securities that are admissible as bids. While our environment is more general, our goal is comparatively modest in that we simply aim to characterize implementability (and not to derive optimal contracts). Additionally, since we do not focus on auctions, we do not need the space of admissible contingent contracts to be ranked - securities are completely ordered and better securities provide a higher expected payoff to the seller irrespective of bidder type. This restriction is required in security auctions to ensure that a winner can be declared based on the bids but before the payoff is realized. The linear contracts we consider cannot not be ranked ex-ante, and hence, are explicitly prohibited in the security auctions literature.

Perhaps one of the reasons that contingent contracts have received limited attention is because of an observation of Crémer (1987). He argued that in a security auction, the principal could only choose to sell a very small share of the future profit. By offering a very low share of the ex-post payoff to the agents, the principal can make the information rents negligible. In other words, the principal can always get arbitrarily close to extracting full surplus. Of course, while this is a sound theoretical argument, it is seldom observed in real world for a number of reasons. For instance, the principal may be liquidity constrained and hence, may be unable to finance the necessary upfront payment to buy the agent. In other cases, as DeMarzo et al. (2005) argue, the agents may have to make noncontractible, fixed, costly investments in order for profits to be realized. If the ex-post payoffs offered by the contingent contract are too low, the agent may choose to just accept the upfront payment and not to undertake the investment. Alternatively, agents may have type dependent or even private outside options and must be offered a higher payoff by the contingent contract (Ekmekci et al., 2014). In practice, environments which feature contingent contracts often have various such legal or practical restrictions which may prevent the principle from extracting all the surplus. For these applications, our characterization of incentive compatibility is an important first step which can help in the derivation of optimal contracts.

This paper is also related to the literature on dominant strategy implementation with transfers - a seminal paper is Rochet (1987). Perhaps the closest paper to ours in this literature is Rahman (2011) who characterizes implementation in an environment where the principal can observe and condition the mechanism on a noisy signal which is correlated with the agent's type. The signal in his model depends only on the type and not on the allocation, and, further, both the scf and the payments are functions both of the signal and the agent's report. By contrast, in our setting, the scf depends only on the reports, but we consider contingent contracts, whereas he restricts attention to transfers.

2 The Deterministic Model

There is a set of agents $N := \{1, \dots, n\}$ who face a mechanism designer (principal). The set of alternatives is A. For ease of exposition, we begin by examining a deterministic model and the majority of the analysis in the paper will be conducted in this framework. Here, the type of an agent i is given by a map $v_i : A \to \mathbb{R}$ and V_i denotes the set of all possible types of agent i. Using the standard notation, $V := V_1 \times \ldots \times V_n$ denotes the set of types of all the agents and $V_{-i} := \prod_{j \neq i} V_j$ is the set of types of all agents except i. In this deterministic environment, the ex-post payoff of agent i with type v_i for an alternative a is given by $v_i(a)$, and is observed by both the agent and the mechanism designer.[2] For notational simplicity, we assume that there are no two distinct types v_i, v_i' such that $v_i(a) = v_i'(a)$ for all $a \in A$.

In Section 4, we describe the general model with uncertainty. There, the ex-post payoff of each agent is a random variable, the distribution of which depends on his type and the alternative chosen. At the interim stage (that is, after the type is realized and before an

[2]We use the term 'payoff' to distinguish this from the final 'payoff' that the contract awards.

alternative is chosen), this ex-post payoff is not known to both the mechanism designer and the agents.

A **social choice function (scf)** is a map $f : V \to A$. This map specifies the chosen alternative for every reported profile of types.

The fundamental difference separating our model from the standard mechanism design setting is that the ex-post payoff of every agent is *contractible*. A commonly observed contract which has this feature is a **linear contract**. A linear contract for agent i consists of two mappings, a royalty (or tax) rule $r_i : V \to (0, 1]$ and a transfer rule $t_i : V \to \mathbb{R}$. A **linear mechanism** $(f, (r_1, t_1), \ldots, (r_n, t_n))$ consists of a linear contract (r_i, t_i) for each agent i and an scf f. The payoff assigned to agent i by such a linear mechanism is

$$r_i(v'_i, v'_{-i})v_i(f(v'_i, v'_{-i})) - t_i(v'_i, v'_{-i}),$$

if his true type is v_i and the profile of reported types is (v'_i, v'_{-i}). In words, a linear contract specifies a transfer amount and a fraction of the payoff to be shared. Notice that we do not allow $r_i(v_i, v_{-i}) = 0$ for any profile of types (v_i, v_{-i}). The main reason we impose this restriction is to prevent the principal from "buying" the agents, thereby making them indifferent amongst reports and trivializing the implementation problem.

A special case of the linear mechanism is the standard **quasi-linear mechanism** $(f, t_1, \ldots, t_n)$, in which the contracts just specify transfers, and where $r_i(\cdot) = 1$ for all i. The payoff assigned to agent i by such a quasi-linear mechanism is $v_i(f(v'_i, v'_{-i})) - t_i(v'_i, v'_{-i})$ if the agent's true type is v_i and the profile of reported types is (v'_i, v'_{-i}).

An important aspect of linear contracts is that the payoff awarded by the contract is increasing in the realized payoff $v_i(\cdot)$ of the agent since the r_i's are restricted to being positive. We now define a general nonlinear class of contracts which satisfy this property. A **contingent contract** of agent i is a map $s_i : \mathbb{R} \times V \to \mathbb{R}$ which is strictly increasing in the first argument. A contingent contract of agent i assigns a payoff to him for every realized ex-post payoff and for every profile of reported types. A **contingent mechanism** is $(f, s_1, \ldots, s_n)$, where f is an scf and $(s_1, \ldots, s_n)$ are the contingent contracts of the agents. The payoff assigned to agent i by such a contingent mechanism is $s_i(v_i(f(v'_i, v'_{-i})), v'_i, v'_{-i})$, if his true type is v_i and the profile of reported types is (v'_i, v'_{-i}). Note that, since s_i is strictly increasing in the first argument, the assigned payoff by a contingent contract is strictly larger for greater realized payoffs. A linear contract is a special case of a contingent contract.

While the contingent contracts we consider are very general and model many real world contracts, they are with loss of generality. Requiring s_i to be strictly increasing in the first argument is not completely innocuous as it rules out certain commonly used contracts which are weakly increasing such as call options and convertible debt. Again, this assumption is made is to prevent the principal from making agents indifferent amongst reports (for instance, by buying the agents). Additionally, notice that we do not allow the payoff to agent i from the contingent contract to depend on the realized payoffs of the other agents but only on their announced types. This is true in most real world contingent contracts and, to the best of our knowledge, this simplifying assumption is made in all of the papers in the literature.

Most importantly, in this deterministic version of our framework, the monotonicity restriction may prevent the principal from punishing detectable misreports from the agent. Here, the realized payoff may reveal the true type of the agent and thus, in principle, contracts can be written which impose large punishments whenever misreports are detected. Such punishments may not be possible using a contingent contract as the monotonicity requirement will then impose a restriction on the payoffs that the contract can offer other agents. That said, we should point out that this deterministic version of our model is merely for expositional purposes and in the general version of our model with uncertainty (described in Section 4), realized payoffs do not generally reveal types.

We now define the notion of dominant strategy implementation that we use.

Definition 1 *An scf f is* **implementable by a linear contract** *in dominant strategies if there exist linear contracts $((r_1, t_1), \dots, (r_n, t_n))$ such that $\forall i \in N$, $\forall v_{-i} \in V_{-i}$,*

$$r_i(v_i, v_{-i})v_i(f(v_i, v_{-i})) - t_i(v_i, v_{-i}) \geq r_i(v'_i, v_{-i})v_i(f(v'_i, v_{-i})) - t_i(v'_i, v_{-i}) \qquad \forall\ v_i, v'_i \in V_i.$$

Then, we say that the linear mechanism $(f, (r_1, t_1), \dots, (r_n, t_n))$ is incentive compatible.

The notion of implementation with contingent contracts can be defined analogously.

Definition 2 *An scf f is* **implementable** *(by a contingent contract) in dominant strategies if there exist contingent contracts $(s_1, \dots, s_n)$ such that $\forall i \in N$, $\forall v_{-i} \in V_{-i}$,*

$$s_i(v_i(f(v_i, v_{-i})), v_i, v_{-i}) \geq s_i(v_i(f(v'_i, v_{-i})), v'_i, v_{-i}) \qquad \forall\ v_i, v'_i \in V_i.$$

Then, we say that the contingent mechanism $(f, s_1, \dots, s_n)$ is incentive compatible.

3 Characterizing Implementability with Linear Contracts

In this section, we present our main characterization result: Every implementable scf can also be implemented by a linear contract. We also give a qualitative description of implementable scfs. To provide intuition on the role played by payoff sharing in expanding the set of implementable outcomes, we begin by presenting an example of Bikhchandani et al. (2006) (see their supplemental material) of an scf which cannot be implemented by transfers but can be implemented by linear contracts.

EXAMPLE **1** Suppose there are two agents with the type space given by $V_1 = \{v_1^1, v_1^2\}$, $V_2 = \{v_2\}$ and the set of alternatives is $A = \{a^1, a^2\}$. The payoffs are given as follows:

	v_1^1	v_1^2	v_2
a^1	3	1	4
a^2	5	2	2

Consider the Rawlsian scf: $f(v_1^i, v_2) = \text{argmax}_{a \in A} \min\{v_1^i(a), v_2(a)\} = a^i$ for $i \in \{1, 2\}$. If f was implementable by transfers, then incentive compatibility for agent 1 would imply

$$v_1^i(f(v_1^i, v_2)) - t_1(v_1^i, v_2) \geq v_1^i(f(v_1^{i'}, v_2)) - t_1(v_1^{i'}, v_2) \text{ for } i, i' \in \{1, 2\}.$$

Summing both constraints, we get $3+2 \geq 5+1$, contradicting implementability by transfers. Intuitively, to prevent v_1^1 from misreporting as v_1^2, the transfer from the latter report should be at least 2 more than the former as v_1^1 gets a higher payoff from a^2. But for any such transfers, type v_1^2 would prefer to report as v_1^1 as there would be a saving of least 2 in transfers and a loss of only 1 from payoff from the worse alternative a^1.

Instead, consider the following linear contract for agent 1:

$$r_1(v_1^1, v_2) = 1,\ t_1(v_1^1, v_2) = 0 \text{ and } r_1(v_1^2, v_2) = \frac{1}{2},\ t_1(v_1^2, v_2) = 0.$$

This is incentive compatible as $r_1(v_1^1, v_2)v_1^1(f(v_1^1)) - t_1(v_1^1, v_2) = 3 > \frac{5}{2} = r_1(v_1^2, v_2)v_1^1(f(v_1^2)) - t_2(v_1^2, v_2)$ and $r_1(v_1^2, v_2)v_2^1(f(v_1^2)) - t_1(v_1^2, v_2) = 1 \geq 1 = r_1(v_1^1, v_2)v_2^1(f(v_1^1)) - t_2(v_1^1, v_2)$.

Here, by keeping half of the payoff of type v_1^2 agent, incentive compatibility can be achieved even without additional transfers. Doing so, makes the agent with type v_1^2 indifferent and makes the payoff from truthtelling strictly higher for agent with type v_1^1. In general, however, a combination of royalty rates and transfers are required for implementation.

3.1 A Foundation for Linear Contracts

We now characterize the set of scfs that are implementable by linear contracts and use this to show our equivalence result. First, we provide a simple necessary condition for implementability.

Given an scf f, for every $i \in N$ and for every $v_{-i} \in V_{-i}$, we define two binary relations $\succeq^f_{v_{-i}}$ and $\succ^f_{v_{-i}}$ on V_i as follows. For notational convenience, we write $\succeq^f \equiv \succeq^f_{v_{-i}}$ and $\succ^f \equiv \succ^f_{v_{-i}}$; the dependence on v_{-i} is implicitly implied. Fix an $i \in N$ and $v_{-i} \in V_{-i}$. For any, $v_i, v_i' \in V_i$, we define

$$v_i' \succeq^f v_i \quad \text{if} \quad v_i'(f(v_i, v_{-i})) \geq v_i(f(v_i, v_{-i})).$$

Further, for any $v_i, v_i' \in V_i$, we define $v_i' \succ^f v_i$ if $v_i'(f(v_i, v_{-i})) > v_i(f(v_i, v_{-i}))$.

A few comments about these binary relations are in order. In words, $v_i' \succeq^f v_i$ if the type v_i' gets a higher payoff than type v_i from the alternative chosen by the scf f for the latter type. Clearly, the relation $\succeq^f$ is reflexive. However, note that the relation is neither antisymmetric, complete nor transitive. It is entirely possible that for types $v_i' \neq v_i$, $v_i'(f(v_i, v_{-i})) \geq v_i(f(v_i, v_{-i}))$ and $v_i(f(v_i', v_{-i})) \geq v_i'(f(v_i', v_{-i}))$ both hold simultaneously (even with either or both of the inequalities being strict) which implies that $\succeq^f$ need not be antisymmetric. Similarly, $\succeq^f$ may neither be complete nor transitive.

Definition 3 *An scf f is* **2-acyclic** *if for all $i \in N$, for all $v_{-i} \in V_{-i}$, and for every pair of types $v_i, v_i' \in V_i$ with $v_i \succeq^f v_i'$, we have $v_i' \not\succ^f v_i$.*

An scf f is **acyclic** *if for all $i \in N$, for all $v_{-i} \in V_{-i}$, and for every sequence of types $v_i^1, \ldots, v_i^k \in V_i$ with $v_i^1 \succeq^f \ldots \succeq^f v_i^k$, we have $v_i^k \not\succ^f v_i^1$.*

In words, f is 2-acyclic if there does not exist a cycle of two types in the relation $\succeq^f$ (with at least one direction being strict). More generally, f is acyclic if there do not exist such cycles of any finite length. The simple lemma below shows that this condition is necessary for implementability.[3]

Lemma 1 *If an scf is implementable, it is acyclic.*

Proof: Let f be an scf that is implementable by contingent contracts $(s_1, \ldots, s_n)$. Fix $v_{-i} \in V_{-i}$ and consider a sequence of types $v_i^1, \ldots, v_i^k \in V_i$ for agent i, such that $v_i^1 \succeq^f \ldots \succeq^f v_i^k$. Hence, $v_i^j(f(v_i^{j+1}, v_{-i})) \geq v_i^{j+1}(f(v_i^{j+1}, v_{-i}))$ for all $j \in \{1, \ldots, k-1\}$. This implies that

$$s_i(v_i^j(f(v_i^j, v_{-i})), v_i^j, v_{-i}) \geq s_i(v_i^j(f(v_i^{j+1}, v_{-i})), v_i^{j+1}, v_{-i}) \geq s_i(v_i^{j+1}(f(v_i^{j+1}), v_{-i}), v_i^{j+1}, v_{-i}).$$

where the first inequality follows from the incentive compatibility of s_i and the second inequality follows from monotonicity of s_i in the first argument. Applying this inequality sequentially over $j \in \{1, \ldots, k-1\}$, we get

$$s_i(v_i^1(f(v_i^1, v_{-i})), v_i^1, v_{-i}) \geq s_i(v_i^k(f(v_i^k, v_{-i})), v_i^k, v_{-i}) \geq s_i(v_i^k(f(v_i^1, v_{-i})), v_i^1, v_{-i}),$$

where the last inequality follows from incentive compatibility. But monotonicity of s_i implies that $v_i^1(f(v_i^1, v_{-i})) \geq v_i^k(f(v_i^1, v_{-i}))$. Hence, $v_i^k \not\succ v_i^1$, which implies that f is acyclic. ■

The following theorem shows that for finite type spaces, this condition is also sufficient for implementation. Importantly, acyclicity is also sufficient for implementation by a linear contract. The proof is in the Appendix.

---

[3] In contrast to our notion of acyclicity over types, Rochet (1987) described an acyclicity condition over alternatives which is necessary (but not sufficient) for implementation with transfers. His condition is neither necessary nor sufficient for implementation with contingent contracts.

Theorem 1 *Suppose the type space is finite. Then, an scf is implementable if and only if it is acyclic. Moreover, every implementable scf can be implemented by a linear contract.*

Remark (i). For every acyclic scf, the proof explicitly constructs a linear contract that implements it. Under a mild condition on the type space, we show that a linear contract can be constructed such that the resulting mechanism is individually rational and the transfer of each agent is non-negative (see the remark immediately following the proof). Further, since each r_i lies in $(0, 1]$, the planner neither needs to make payments to nor take away large amount of payoff from the agents.[4]

Remark (ii). Finite types are required for the equivalence between implementability by contingent and linear contracts. The supplement (Deb and Mishra, 2014) contains an example of a single agent with a countably infinite type space where this equivalence does not hold.

Remark (iii). Theorem 1 uncovers a parallel with Afriat's theorem (Afriat, 1967) of revealed preference in consumer theory.[5] The acyclicity condition we use to characterize implementability is analogous to the Generalized Axiom of Revealed Preference (Varian, 1982), which is a necessary and sufficient condition for a finite price consumption data set to be rationalized by a utility maximizing consumer. Further, Afriat shows that a data set can be rationalized by a utility function if and only if it can be rationalized by a concave payoff function. Analogously, we show that acyclicity is necessary and sufficient for implementability using either contingent or linear contingent contracts. By contrast, implementability by transfers is characterized by cycle monotonicity which is stronger than acyclicity.[6]

3.2 A Qualitative Description of Implementable SCFs

In this section, we qualitatively describe the set of implementable scfs. For this, we need to introduce some new notation. Given a type profile $v \equiv (v_1, \ldots, v_n) \in V$, we can define a vector $v^a \equiv (v_1(a), \ldots, v_n(a)) \in \mathbb{R}^n$ for each alternative $a \in A$. This is the payoff vector of the agents corresponding to alternative a. Given a type space V, it induces a set of permissible payoff vectors for each alternative. We will denote the set of payoff vectors v^a of alternative a as $\mathcal{U}^a$.

[4] This is true even for standard quasi-linear mechanisms - every implementable scf can be implemented (under reasonable conditions) using individually rational and non-negative transfers (Kos and Messner, 2013).

[5] Beginning with (Rochet, 1987), there have been informal analogies made between these two problems.

[6] Implementability of an scf by transfers can be considered to be analogous to rationalizability of choice data by quasilinear utility functions (Brown and Calsamiglia, 2007).

We will now define the notion of an aggregate payoff function. Define the following set

$$\mathcal{X} := \{(a, x) : a \in A, x \in \mathcal{U}^a\}.$$

An **aggregate payoff function** is a map $W : \mathcal{X} \to \mathbb{R}$. An aggregate payoff function W is **monotone** if for every $a \in A$ and every $x, y \in \mathcal{U}^a$ such that $y \geq x$, we have $W(a, y) \geq W(a, x)$.[7]

DEFINITION **4** *A social choice function f is an* **aggregate payoff maximizer (APM)** *if there exists a monotone aggregate payoff function $W : \mathcal{X} \to \mathbb{R}$ such that at every profile $v \in V$, we have*

$$f(v) \in \operatorname*{argmax}_{a \in A} W(a, v^a).$$

Further, an APM f satisfies consistent tie-breaking if there exists a strict linear order P on A such that at every profile $v \in V$, $f(v)$ is the maximum alternative in the set $\{a \in A : W(a, v^a) \geq W(b, v^b) \ \forall\ b \in A\}$ with respect to the strict linear order P.

This class of scfs include a number of commonly used social welfare functions. An example is the class of affine maximizers. An scf f is an affine maximizer if there exist non-negative weights $(\gamma_1, \ldots, \gamma_n) \in \mathbb{R}^n_+ \setminus \{0\}$ and a map $\kappa : A \to \mathbb{R}$ such that for all $v \in V$,

$$f(v) \in \operatorname*{argmax}_{a \in A} \left[\sum_{i \in N} \gamma_i v_i(a) - \kappa(a) \right].$$

Another example is the Max-min or Rawlsian scf. An scf f is a max-min scf if for all $v \in V$,

$$f(v) \in \operatorname*{argmax}_{a \in A} \min_{i \in N} v_i(a).$$

As Example 1 demonstrated, the Rawlsian scf is not implementable using transfers in general.

The next result shows that an APM with consistent tie-breaking is acyclic and hence implementable. The proof demonstrates that acyclicity is easy to apply.

THEOREM **2** *In a finite type space, every APM with consistent tie-breaking is implementable.*

Proof: We will show that if f is an APM satisfying consistent tie-breaking, then it is acyclic. By Theorem 1, we will be done. Let P be the linear order on the set of alternatives that is used to consistently break ties in f. Further, let W be a monotone aggregate payoff function such that $f(v) \in \operatorname{argmax}_{a \in A} W(a, v^a)$ for every $v \in V$.

[7]For any $x, y \in \mathbb{R}^n$, if $x_i \geq y_i$ for all $i \in N$, we write $x \geq y$.

Fix an agent $i \in N$ and type profile of other agents at v_{-i}. Consider a sequence of types $v_i^1 \succeq^f \ldots \succeq^f v_i^k$. Pick any $j \in \{1, \ldots, k-1\}$. Let $f(v_i^j, v_{-i}) = a_j$ and $f(v_i^{j+1}, v_{-i}) = a_{j+1}$. Since $v_i^j \succeq^f v_i^{j+1}$, we have $v_i^j(a_{j+1}) \geq v_i^{j+1}(a_{j+1})$. Denote the payoff vector of any alternative c in type profile (v_i^j, v_{-i}) as $v^{j,c}$. Since $f(v_i^j, v_{-i}) = a_j$, we have $W(a_j, v^{j,a_j}) \geq W(a_{j+1}, v^{j,a_{j+1}})$ and monotonicity of W gives $W(a_{j+1}, v^{j,a_{j+1}}) \geq W(a_{j+1}, v^{(j+1),a_{j+1}})$. Combining these inequalities, we get

$$W(a_j, v^{j,a_j}) \geq W(a_{j+1}, v^{j,a_{j+1}}) \geq W(a_{j+1}, v^{(j+1),a_{j+1}}).$$

Using it over all $j \in \{1, \ldots, k-1\}$, we get that

$$W(a_1, v^{1,a_1}) \geq W(a_2, v^{1,a_2}) \geq W(a_2, v^{2,a_2}) \geq \ldots \geq \ldots \geq W(a_k, v^{k-1,a_k}) \geq W(a_k, v^{k,a_k}).$$

Since $f(v_i^k, v_{-i}) = a_k$, we know that $W(a_k, v^{k,a_k}) \geq W(a_1, v^{k,a_1})$. Hence, we get

$$W(a_1, v^{1,a_1}) \geq W(a_1, v^{k,a_1}). \tag{1}$$

Now, assume for contradiction that $v_i^k \succ^f v_i^1$. So, $v_i^k(a_1) > v_i^1(a_1)$. By monotonicity of W, we have $W(a_1, v^{k,a_1}) \geq W(a_1, v^{1,a_1})$. Using Inequality (1), we get

$$W(a_1, v^{1,a_1}) = W(a_2, v^{1,a_2}) = W(a_2, v^{2,a_2}) = \ldots = W(a_k, v^{k,a_k}) = W(a_1, v^{k,a_1}) = W(a_1, v^{1,a_1}).$$

Now, pick any $j \in \{1, \ldots, k-1\}$. Since $W(a_j, v^{j,a_j}) = W(a_{j+1}, v^{j,a_{j+1}})$, by consistent tie-breaking, it must be that either $a_j = a_{j+1}$ or $a_j P a_{j+1}$. Using it for all $j \in \{1, \ldots, k-1\}$, we see that either $a_1 = a_2 = \ldots = a_k$ or $a_1 P a_k$. But $W(a_k, v^{k,a_k}) = W(a_1, v^{k,a_1})$ implies that $a_1 P a_k$ is not possible. Hence, $a_1 = a_2 = \ldots = a_k = a$ for some $a \in A$. But this implies that $v_i^1(a) \geq v_i^2(a) \geq \ldots \geq v_i^k(a)$, and this contradicts that $v_i^k \succ^f v_i^1$. ■

Under additional conditions, we can show the converse of Theorem 2. We require the following richness in type space.

Definition 5 *The type space V is* **rich** *if the set of profiles of payoff vectors is $\mathcal{U}^a \times \mathcal{U}^b \times \ldots$*

This richness condition requires that every combination of payoff vectors is a feasible type profile. For instance, if v^a and v'^a are two payoff vectors corresponding to alternative a in $\mathcal{U}^a$ and (v^a, v^{-a}) is a profile of payoff vectors at a type profile, then the profile of payoff vectors (v'^a, v^{-a}) must correspond to a valid type profile in the type space V. Let $\mathcal{U} := \mathcal{U}^a \times \mathcal{U}^b \times \ldots$.

We now impose an independence condition on the scfs. It is in the spirit of binary independence used in the social choice theory literature (d'Aspremont and Gevers, 2002) from where we borrow the terminology.

Definition 6 *An scf f satisfies* **binary independence** *if for every distinct pair of alternatives $a, b \in A$ and every $v, v' \in V$ such that $v^a = v'^a, v^b = v'^b$, $f(v) = a$ implies that $f(v') \neq b$.*

Binary independence requires that if a is chosen over b as the outcome by an scf at a type profile, then b cannot be chosen at a different type profile in which the payoff vectors corresponding to a and b are not changed. In other words, the scf must evaluate a and b at any type profile independent of payoff vectors of other alternatives. The implication of binary independence is well understood in social choice theory (d'Aspremont and Gevers, 2002). It helps us to break ties in a consistent manner. We now show that the APMs with consistent tie-breaking are the only implementable scfs under binary independence. The proof of this result is in the Appendix.

Theorem 3 *In a finite and rich type space, the following are equivalent for an scf f.*

1. *f is an aggregate payoff maximizer with consistent tie-breaking.*
2. *f is implementable and satisfies binary independence.*
3. *f is 2-acyclic and satisfies binary independence.*

Notice that Theorem 3 also shows that, under these additional conditions, the significantly weaker 2-acyclicity is sufficient for implementation. It is worth comparing Theorem 3 to a similar characterization for the quasi-linear mechanisms. Roberts (1979) showed that affine maximizers are the only implementable scfs using quasi-linear mechanisms. Though Theorem 3 can be viewed as counterpart of that result in the contingent contract environment, there are significant differences. While we require a finite and rich type space with binary independence, Roberts required a finite set of alternatives with at least three alternatives, the type space to be the whole of $\mathbb{R}^{|A|}$, and the scf to be onto.

3.3 Discussion on Payoffs

While, we have provided a foundation for linear contracts in terms of implementability, a natural question to ask is whether the payoffs from every contingent mechanism can also be achieved by a linear mechanism. More precisely, given an scf f and contingent contracts $(s_1, \ldots, s_n)$ that implement it, we ask if there exist linear contracts $((r_1, t_1), \ldots, (r_n, t_n))$ that implement f such that

$$s_i(v_i(f(v_i, v_{-i})), v_i, v_{-i}) = r_i(v_i, v_{-i})v_i(f(v_i, v_{-i})) - t_i(v_i, v_{-i}) \text{ for all } i \in N, v_i \in V_i, v_{-i} \in V_{-i}.$$

Note that this requirement is only for payoffs on the equilibrium path. The following single agent example shows that this payoff equivalence does not hold.[8]

EXAMPLE **2** Consider a single agent with type space $V_1 := \{v_1^1, v_1^2, v_1^3\}$ and the set of alternatives $A := \{a^1, a^2, a^3\}$. The payoffs are given below.

	v_1^1	v_1^2	v_1^3
a^1	30	20	10
a^2	30	20	10
a^3	20	10	10

The scf $f(v_1^j) = a^j$ for $j \in \{1, 2, 3\}$ is implemented by the following contingent contract:

$$s_1(30, v_1^1) = 20, s_1(20, v_1^1) = 5, s_1(10, v_1^1) = 1,$$
$$s_1(30, v_1^2) = 16, s_1(20, v_1^2) = 15, s_1(10, v_1^2) = 1,$$
$$s_1(20, v_1^3) = 10, s_1(10, v_1^3) = 5.$$

Now, suppose there is a payoff equivalent linear contract (r_1, t_1) that implements f. Then,

$$r_1(v_1)v_1(f(v_1)) - t_1(v_1) = s_1(v_1(f(v_1)), v_1) \text{ for all } v_1 \in V_1.$$

Incentive compatibility of the linear mechanism would then imply that for all v_1, v_1',

$$s_1(v_1'(f(v_1')), v_1') - s_1(v_1(f(v_1)), v_1) \le r_1(v_1')[v_1'(f(v_1')) - v_1(f(v_1'))]. \tag{2}$$

Taking $v_1' = v_1^2$ and $v_1 = v_1^3$ in Inequality (2), we get $r_1(v_1^2) \ge 1$. Taking $v_1' = v_1^2$ and $v_1 = v_1^1$ in Inequality (2), we get $r_1(v_1^2) \le \frac{1}{2}$, which is a contradiction.

Though payoff equivalence does not hold between linear mechanisms and non-linear contingent mechanisms, we can show that linear contracts can expand the set of payoffs achievable using quasilinear mechanisms. We do this by providing a simple and powerful application of linear contracts. It is well known that the efficient scf f^*, defined as $f^*(v) \in \text{argmax}_{a\in A} \sum_{i\in N} v_i(a)$, can be implemented using the VCG mechanisms, but they are not budget-balanced in many environments (Walker, 1980). However, it is easy to verify that the following simple linear contract,

$$r_i^*(v) = \frac{1}{n}, \quad t_i^*(v) = -\frac{1}{n}\sum_{j\neq i} v_j(f^*(v)) \;\; \forall\, v \in V, \;\forall\, i \in N,$$

[8] The failure of payoff equivalence is not driven by finite type space restriction and it is easy to construct similar examples with a continuum of types. Further, the usual revenue/payoff equivalence in quasi-linear environments requires that two transfers implementing the same scf must differ in payoffs by a constant. However, the payoff equivalence that we seek is across two *classes* of contracts implementing the same scf.

implements f^* and the resulting linear mechanism awards each agent an equal $1/n$ share of the total social surplus. This implies that this mechanism is budget-balanced. In other words, with linear contracts, the principal can always achieve efficiency without transferring money to (or taking money from) the agents. We leave the important question of characterizing the set of payoffs achievable using linear contracts for future research.

4 The General Model with Uncertainty

In this section, we present the general model with uncertainty and discuss how the results in the previous section extend to this environment. For this, we will need some additional notation. The type v_i of the agent now determines the distribution of the ex-post payoff that an agent receives from an alternative a. We denote by u_i, the random variable for agent i corresponding to the ex-post payoff. At the interim stage (that is, after realization of the type and before an alternative is chosen), this payoff is not known to the agent and the mechanism designer. It is assumed that when agent i has type v_i, his ex-post payoff u_i from alternative a is drawn from $\mathbb{R}$ with cumulative distribution $G^a_{v_i}$ which depends both on the true type and the alternative. Note that since the payoff is a random variable, its realization need not reveal the type of the agent.[9] In a minor abuse of notation, we use $v_i(a)$ to denote the expected payoff from alternative a or $v_i(a) = \int_{\mathbb{R}} u_i dG^a_{v_i}(u_i)$.

We will impose the following restriction on the distribution of payoffs.

Definition **7** *The distributions of payoffs are ordered by first order stochastic dominance or simply* **ordered** *if for all i, $v_i, v_i' \in V_i$ and for all $a \in A$, we have*

$$\text{either } G^a_{v_i} \succeq_{FOSD} G^a_{v_i'} \text{ or } G^a_{v_i'} \succeq_{FOSD} G^a_{v_i},$$

where $\succeq_{FOSD}$ is the first-order stochastic dominance relation.

This ordering requirement says that for every agent i and every alternative $a \in A$, the types in V_i can be ex-ante ordered using the $\succeq_{FOSD}$ relation. Note that this does not imply that the ordering of types has to be the same across the different alternatives. To the best of our knowledge, most of the theoretical work on mechanism design with contingent contracts

[9] Of course, if the principal knew the prior distribution over the agents' types, the realized payoff would allow him to update the prior. By contrast, if the principal does not know the type distribution, he will not be able to make inference (dominant strategy implementation is appropriate for these cases). That said, we allow the supports of the distributions of payoffs to vary over different alternatives. Hence, even without prior knowledge of how the types are distributed, there may be certain realizations of payoff from which the principal can back out the type of the agent.

requires this assumption.[10] Importantly, the deterministic environment (which corresponds to the distributions being degenerate) we have studied in the previous sections is ordered in the above sense.

Finally, as in the deterministic case, we assume for notation simplicity that there are no duplicate types. In other words, for all agents $i \in N$, there are no two types $v_i, v_i' \in V_i$ such that $v_i(a) = v_i'(a)$ for all $a \in A$.

We can now define an scf and the contracts analogously to the deterministic environment. As before, an scf is a mapping $f : V \to A$. Linear contracts are defined identically and consist of functions $r_i : V \to (0, 1]$ and $t_i : V \to \mathbb{R}$ for each agent i. Similarly, contingent contracts are mappings $s_i : \mathbb{R} \times V \to \mathbb{R}$ for each agent i which is strictly increasing in the first argument.

Dominant strategy implementation can be adapted in a natural way. Agents now compute their *expected payoff* before reporting their types. The definition of implementability by linear contracts looks identical to the deterministic case with the only difference being that the $v_i(\cdot)$ in the incentive compatibility constraints now denotes the expected payoff.

Definition 8 *An scf f is* **implementable** *if there exist contingent contracts $(s_1, \ldots, s_n)$ such that , $\forall\ i$, $v_i, v_i' \in V_i$ and $v_{-i} \in V_{-i}$, we have*

$$\int_{\mathbb{R}} s_i(u_i, v_i, v_{-i}) dG_{v_i}^{f(v_i, v_{-i})}(u_i) \geq \int_{\mathbb{R}} s_i(u_i, v_i', v_{-i}) dG_{v_i}^{f(v_i', v_{-i})}(u_i).$$

In this case, we say that the contingent contracts $(s_1, \ldots, s_n)$ implement f and the contingent mechanism $(f, s_1, \ldots, s_n)$ is incentive compatible.

Ordering of the distributions is essential because acyclicity only characterizes implementability under this condition. Note that, the definition of acyclicity remains unchanged with, once again, the difference being that the $v_i(\cdot)$'s used to define the relations $\succeq^f$ and $\succ^f$ are expected payoffs. Note also that since the type space is assumed to be ordered $v_i(a) \geq (>) v_i'(a)$ is equivalent to $G_{v_i}^a \succeq_{FOSD} (\succ_{FOSD}) G_{v_i'}^a$. Finally, observe that if the type space is ordered, acyclicity remains necessary for implementation. This is because for ordered types $v_i', v_i \in V_i$, the following holds

$$v_i' \succeq^f v_i \quad \Longrightarrow \quad \int_{\mathbb{R}} s_i(u_i, v_i, v_{-i}) dG_{v_i'}^{f(v_i, v_{-i})}(u_i) \geq \int_{\mathbb{R}} s_i(u_i, v_i, v_{-i}) dG_{v_i}^{f(v_i, v_{-i})}(u_i),$$

where the inequality follows from the monotonicity of s_i in u_i and the fact that $G_{v_i'}^{f(v_i, v_{-i})}$ first order stochastically dominates $G_{v_i}^{f(v_i, v_{-i})}$. As in the proof of Lemma 1, this combined with incentive compatibility ensures that every implementable scf must be acyclic.

[10] Often, the stronger assumption of affiliation (Milgrom and Weber, 1982) is made instead (Skrzypacz, 2013).

All of our results generalize to this general environment with uncertainty. The characterization result, Theorem 1, holds verbatim with the adjusted definition of acyclicity. A natural way to define aggregate payoff maximizers is in terms of expected payoffs and, with this definition, Theorem 2 continues to hold as stated. The definition of payoff vector of the agents corresponding to a given alternative a, $\mathcal{U}^a$, will remain the same with the $v_i(\cdot)$'s now being expected payoffs. Theorem 3, also holds verbatim with the adjusted definitions of 2-acyclicity, richness and binary independence.

5 Extensions

We end the paper by discussing some extensions to the results in the paper. All the results that we describe below can be found in the supplement (Deb and Mishra, 2014).

An assumption in our model is that the entire realized payoff of the agents is contractible. While this is appropriate in many settings, it is a strong assumption for others. We can extend Theorem 1 to an environment where the realized payoff is in two parts – contractible and noncontractible. We show that as long as both are comonotone, the result will continue to hold. Of course, an important extension for future work is to examine environments in which these are not comonotone where Theorem 1 does not hold in general.

We can show that the equivalence of implementation between linear and contingent contracts holds in uncountable type spaces under additional smoothness conditions. However, the smoothness we require for this result is often absent in many practical applications such as auctions. We hope to conduct a more formal analysis of uncountable type spaces in the future. Here, a natural question is: When the equivalence of linear contracts and contingent contracts in Theorem 1 fails, is there is a different class of simple (nonlinear) contracts which are sufficient for implementation?

Although we verified acyclicity for aggregate payoff maximizers, it may, in principle, be difficult to check for other applications. This is because it requires checking for the absence of cycles of all finite lengths. Theorem 3 helped in this regard by showing that the substantially weaker condition 2-acyclicity is sufficient but only as long as the type space is rich. We can show that 2-acyclicity is sufficient in certain commonly utilized settings even when the type space is not rich – linear one dimensional environments with uncountable types and linear two dimensional environments with countable types.

Another interesting generalization would be to consider interdependent value settings. Here, even the efficient scf cannot generically be implemented using transfers (Jehiel and Moldovanu, 2001). However, Mezzetti (2004) showed that this can be overcome by using two-stage mechanisms that depend on the realized payoffs of agents. We are not aware of work analyzing the implementability of Rawlsian scfs in an interdependent value setting.

Appendix

Proof of Theorem 1

Throughout the proof, we fix an agent i and type profile of other agents at v_{-i}. For notational convenience, we suppress the dependence on v_{-i}. We begin the proof by noting that a consequence of acyclicity is that the type space can be partitioned. A type space V_i can be f-order-partitioned if there exists a partition $(V_i^1, \ldots, V_i^K)$ of the type space V_i such that

P1 for each $j \in \{1, \ldots, K\}$ and for each $v_i, v_i' \in V_i^j$, we have $v_i \not\succ^f v_i'$,

P2 for each $j \in \{1, \ldots, K-1\}$, for each $v_i \in V_i^j$, and for each $v_i' \in (V_i^{j+1} \cup \ldots \cup V_i^K)$, we have $v_i' \not\succeq^f v_i$.

We first show that any acyclic SCF f induces an f-ordered-partition of the type space.

Lemma **2** *Suppose the type space is finite and f is an acyclic SCF. Then, the type space can be f-ordered-partitioned.*

Proof: Let f be an acyclic scf. Consider any non-empty subset $V_i' \subseteq V_i$. A type v_i is maximal in V_i' with respect to $\succ^f$ if there exists no type $v_i' \in V_i'$ such that $v_i' \succ^f v_i$. Denote the set of types that are maximal in V_i' with respect to $\succ^f$ as $\widetilde{V_i'}$. Since f is acyclic, $\succ^f$ is acyclic. Since V_i' is finite, we conclude that $\widetilde{V_i'}$ is non-empty (Sen, 1970). Define

$$M(V_i') := \{v_i \in \widetilde{V_i'} : \; v_i' \not\succeq^f v_i \;\forall\; v_i' \in V_i' \setminus \widetilde{V_i'}\}.$$

We claim that $M(V_i')$ is non-empty. Assume for contradiction that $M(V_i')$ is empty. Choose $v_i^1 \in \widetilde{V_i'}$. Since $M(V_i')$ is empty, there exists $\bar{v}_i{}^1 \in V_i' \setminus \widetilde{V_i'}$ such that $\bar{v}_i{}^1 \succeq^f v_i^1$. Since $\bar{v}_i{}^1 \in V_i' \setminus \widetilde{V_i'}$, there exist a sequence of types $(v_i^2, \ldots, v_i^k)$ such that $v_i^2 \succ^f \ldots \succ^f v_i^k \succ^f \bar{v}_i{}^1 \succeq^f v_i^1$ and $v_i^2 \in \widetilde{V_i'}$. Since $v_i^2 \in \widetilde{V_i'}$ and $M(V_i')$ is empty, there must exist $\bar{v}_i{}^2 \in V_i' \setminus \widetilde{V_i'}$ such that $\bar{v}_i{}^2 \succeq^f v_i^2$. This process can be repeated. Since V_i' is finite, we will get a cycle of types satisfying $v_i \ldots \succeq^f \ldots \succ^f \ldots v_i$. Since f is acyclic, $v_i \not\succeq^f v_i$. But this contradicts the fact that $\succeq^f$ is reflexive. Hence, $M(V_i')$ is non-empty.

We note that for any $v_i, v_i' \in M(V_i')$, we have $v_i \not\succ^f v_i'$. Now, we recursively define the f-ordered partition of V_i. First, we set $V_i^1 := M(V_i)$. Having defined $V_i^1, \ldots, V_i^k$, we define $R^k := V_i \setminus (V_i^1 \cup \ldots \cup V_i^k)$. If $R^k \neq \emptyset$, then define $V_i^{k+1} := M(R^k)$ and repeat. If $R^k = \emptyset$, then $V_i^1, \ldots, V_i^k$ is an f-ordered partition of V_i by construction. ■

A consequence of Lemma 2 is that f satisfies the following property.

Definition 9 *An scf f satisfies* **scaled K-cycle monotonicity**, *where $K \geq 2$ is a positive integer, if there exists $\lambda_i : V_i \to (0, \infty)$ such that for all sequence of types $(v_i^1, \ldots, v_i^k)$ with $k \leq K$, we have*

$$\sum_{j=1}^{k} \lambda_i(v_i^j)\big[v_i^j(f(v_i^j)) - v_i^{j+1}(f(v_i^j))\big] \geq 0, \tag{3}$$

where $v_i^{k+1} \equiv v_i^1$. An scf f is scaled cycle monotone (scm) if it satisfies scaled K-cycle monotonicity for all integers $K \geq 2$. In this case, we say λ_i makes f scm.

To show that f is scm, we construct a λ_i that makes it scm.

Constructing λ_i: We use Lemma 2 to construct the λ_i map recursively. Let f be an acyclic SCF and $(V_i^1, \ldots, V_i^K)$ be the f-ordered-partition according to Lemma 2. First, we set

$$\lambda_i(v_i) = 1 \ \forall \ v_i \in V_i^K. \tag{4}$$

Having defined $\lambda_i(v_i)$ for all $v_i \in (V_i^{k+1} \cup V_i^{k+2} \cup \ldots \cup V_i^K)$, we define $\lambda_i(v_i)$ for all $v_i \in V_i^k$.

Let C be any cycle of types $(v_i^1, \ldots, v_i^q, v_i^1)$ involving types in $(V_i^k \cup V_i^{k+1} \cup \ldots V_i^K)$ with at least one type in V_i^k and at least one type in $(V_i^{k+1} \cup \ldots \cup V_i^K)$. Let $\mathcal{C}$ be the set of all such cycles. For each cycle $C \equiv (v_i^1, \ldots, v_i^q, v_i^{q+1} \equiv v_i^1) \in \mathcal{C}$, [11] define

$$L(C) = \sum_{v_i^j \in C \cap (V_i^{k+1} \cup \ldots \cup V_i^K)} \lambda_i(v_i^j)\big[v_i^j(f(v_i^j)) - v_i^{j+1}(f(v_i^j))\big] \tag{5}$$

$$\ell(C) = \sum_{v_i^j \in C \cap V_i^k} \big[v_i^j(f(v_i^j)) - v_i^{j+1}(f(v_i^j))\big]. \tag{6}$$

We now consider two cases.

Case 1. If $L(C) \geq 0$ for all $C \in \mathcal{C}$, then we set $\lambda_i(v_i) = 1$ for all $v_i \in V_i^k$.

Case 2. If $L(C) < 0$ for some $C \in \mathcal{C}$, we proceed as follows. Since V_i is f-ordered partitioned, for every $v_i \in V_i^k$ and $v_i' \in (V_i^{k+1} \cup \ldots \cup V_i^K)$, we have $v_i' \not\succeq^f v_i$ (Property P1 of f-ordered partition), and hence, $v_i(f(v_i)) - v_i'(f(v_i)) > 0$. Similarly, for every $v_i, v_i' \in V_i^k$, we have $v_i' \not\succ^f v_i$ (Property P2 of f-ordered partition), and hence, $v_i(f(v_i)) - v_i'(f(v_i)) \geq 0$.

Then, for every $C \in \mathcal{C}$, we must have $\ell(C) > 0$ since C involves at least one type from V_i^k and at least one type from $(V_i^{k+1} \cup \ldots \cup V_i^k)$. Now, for every $v_i \in V_i^k$, define

$$\lambda_i(v_i) := \max_{C \in \mathcal{C}: L(C) < 0} \frac{-L(C)}{\ell(C)}. \tag{7}$$

We can thus recursively define the λ_i map.

[11] We will abuse notation to denote the set of types in a cycle C by C also.

Proposition 1 *Suppose V_i is finite. If an scf f is acyclic, then λ_i makes f scm.*

Proof: Suppose f is acyclic. By Lemma 2, V_i can be f-ordered-partitioned. Let the induced partition of V_i be $(V_i^1, \ldots, V_i^K)$ and let λ be defined recursively as before using Equations (4) and (7). Consider any cycle $C \equiv (v_i^1, \ldots, v_i^q, v_i^{q+1} \equiv v_i^1)$. We will show that

$$\sum_{v_i^j \in C} \lambda_i(v_i^j)\big[v_i^j(f(v_i^j)) - v_i^{j+1}(f(v_i^j))\big] \geq 0. \tag{8}$$

If $C \subseteq V_i^K$, then $v_i^j(f(v_i^j)) - v_i^{j+1}(f(v_i^j)) \geq 0$ (by Property P1 above) and $\lambda_i(v_i^j) = \lambda_i(v_i^{j+1})$ for all $v_i^j, v_i^{j+1} \in C$. Hence, Inequality (8) holds.

Now, suppose Inequality (8) is true for all cycles $C \subseteq (V_i^{k+1} \cup \ldots V_i^K)$. Consider a cycle $C \equiv (v_i^1, \ldots, v_i^q, v_i^{q+1} \equiv v_i^1)$ involving types in $(V_i^k \cup \ldots \cup V_i^K)$. If each type in C is in V_i^k, then again $v_i^j(f(v_i^j)) - v_i^{j+1}(f(v_i^j)) \geq 0$ (by Property P1 above) and $\lambda_i(v_i^j) = \lambda_i(v_i^{j+1})$ for all $v_i^j, v_i^{j+1} \in C$. Hence, Inequality (8) holds. By our hypothesis, if all types in C belong to $(V_i^{k+1} \cup \ldots \cup V_i^K)$, then again Inequality (8) holds. So, assume that C is a cycle that involves at least one type from V_i^k and at least one type from $(V_i^{k+1} \cup \ldots \cup V_i^K)$. Let $\lambda_i(v_i) = \mu$ for all $v_i \in V_i^k$. By definition,

$$\sum_{v_i^j \in C} \lambda_i(v_i^j)\big[v_i^j(f(v_i^j)) - v_i^{j+1}(f(v_i^j))\big] = L(C) + \mu\ell(C) \geq 0,$$

where the last inequality followed from the definition of μ (Equation (7)). Hence, Inequality (8) again holds. Proceeding like this inductively, we complete the proof. ■

Using λ_i, we can define our linear contract that implements f. For this, we need to now define the transfers.

Constructing t_i: If λ_i makes f scm, then λ_i satisfies Inequality ((3)) for any cycle of types. Hence, by Rochet-Rockafellar cycle monotonicity characterization (Rochet, 1987; Rockafellar, 1970), there exists a map $W_i : V_i \to \mathbb{R}$ such that

$$W_i(v_i) - W_i(v_i') \leq \lambda_i(v_i)\big[v_i(f(v_i)) - v_i'(f(v_i))\big] \;\forall\; v_i, v_i' \in V_i. \tag{9}$$

The explicit construction of W_i involves construction of a weighted directed graph and finding shortest paths in such a graph - see Vohra (2011). From this, we can define $t_i : V_i \to \mathbb{R}$ as follows.

$$t_i(v_i) = \lambda_i(v_i)v_i(f(v_i)) - W_i(v_i) \;\forall\; v_i \in V_i.$$

Proposition 2 *If λ_i makes f scm, then (λ_i, t_i) is an incentive compatible linear mechanism.*

Proof: Substituting in Inequality (9), we get for all $v_i, v'_i \in V_i$,

$$\lambda_i(v_i)v_i(f(v_i)) - t_i(v_i) - \lambda_i(v'_i)v'_i(f(v'_i)) + t_i(v'_i) \leq \lambda_i(v_i)\big[v_i(f(v_i)) - v'_i(f(v_i))\big]$$

This gives us the desired incentive constraints: for all $v_i, v'_i \in V_i$

$$\lambda_i(v'_i)v'_i(f(v'_i)) - t_i(v'_i) \geq \lambda_i(v_i)v'_i(f(v_i)) - t_i(v_i).$$

■

Remark. Consider a type space V_i and assume that there exists a type $\underline{v}_i \in V_i$ such that $\underline{v}_i(a) = 0$ for all $a \in A$. Further, assume that for every $v_i \in V_i$ and for every $a \in A$, we have $v_i(a) \geq 0$. In this type space, we can show that there is a linear mechanism that will be individually rational and the payments of agents will be non-negative. To see this, the W_i map constructed in the proof can be constructed such that $W_i(\underline{v}_i) = 0$ - this is easily done by translating any W_i map to a new map with $W_i(\underline{v}_i) = 0$. In that case, the net payoff of agent i when his type is v_i is given by

$$\lambda_i(v_i)v_i(f(v_i)) - t_i(v_i) = W_i(v_i) \geq W_i(\underline{v}_i) - \lambda_i(\underline{v}_i)\big[\underline{v}_i(f(\underline{v}_i)) - v_i(f(\underline{v}_i))\big] = \lambda_i(\underline{v}_i)v_i(f(\underline{v}_i)) \geq 0.$$

Similarly, for any v_i,

$$W_i(v_i) \leq W_i(\underline{v}_i) + \lambda_i(v_i)\big[v_i(f(v_i)) - \underline{v}_i(f(v_i))\big] = \lambda_i(v_i)v_i(f(v_i)).$$

Hence, $t_i(v_i) \geq 0$. Finally, note that we can always scale (λ_i, t_i) such that λ_i lies between 0 and 1 while maintaining $t_i(\cdot) \geq 0$ and individual rationality. Hence, there are linear mechanisms in this type space where the payments of agents are non-negative and all agents are individually rational.

Proof of Theorem 3

$1 \Rightarrow 2$ and $2 \Rightarrow 3$. Clearly, an APM with consistent tie-breaking satisfies binary independence and it is implementable by Theorem 2. $2 \Rightarrow 3$ follows from Theorem 1.

$3 \Rightarrow 1$. We do this part of the proof in many steps. Let f be a 2-acyclic scf satisfying binary independence.

Step 1. We show that f satisfies the following positive association property. We say f satisfies **weak positive association (WPA)** if for every pair of type profiles v, v' with $f(v) = a$, $v'_i(a) \geq v_i(a)$ for all $i \in N$, $v'_i(x) = v_i(x)$ for all $x \neq a$, for all $i \in N$, we have $f(v') = a$.

To see this, consider two type profiles v and $(\bar{v}_i, v_{-i})$ with $f(v) = a$, $\bar{v}_i(a) > v_i(a)$, and $\bar{v}_i(x) = v_i(x)$ for all $x \neq a$. Assume for contradiction $f(\bar{v}_i, v_{-i}) = b \neq a$. So, we have $\bar{v}_i(a) > v_i(a)$ and $v_i(b) = \bar{v}_i(b)$, and this contradicts 2-acyclicity of f. By repeatedly applying this argument for all $i \in N$, we get that f satisfies WPA.

Step 2. Let $\bar{A} := \{a \in A :$ there exists $v \in V$ such that $f(v) = a\}$, i.e., $\bar{A}$ is the range of f. Let $\bar{\mathcal{X}} := \{(a, x) \in \mathcal{X} : a \in \bar{A}\}$. Note that since V is finite, $\bar{A}$ (the range of f) is finite. As a result, $\bar{\mathcal{X}}$ is also finite. Now, we define a binary relation $\rhd^f$ on the elements of $\bar{\mathcal{X}}$. For any $(a, x), (b, y) \in \bar{\mathcal{X}}$ with $a \neq b$, we let

$$(a, x) \rhd^f (b, y) \text{ if there exists } v \in V \text{ such that } v^a = x, v^b = y, f(v) = a$$

and for any $(a, x), (a, x + \epsilon) \in \bar{\mathcal{X}}$ with $\epsilon \in \mathbb{R}^n_+$ and $x \neq (x + \epsilon)$, we let

$$(a, x + \epsilon) \rhd^f (a, x).$$

Note that the binary relation is only a partial order. Binary independence immediately implies that $\rhd^f$ is anti-symmetric.

Step 3. We will say that the binary relation $\rhd^f$ satisfies a monotonicity property. Pick distinct $a, b \in \bar{A}$ and $x \in \mathcal{U}^a, y \in \mathcal{U}^b$ such that $(a, x) \rhd^f (b, y)$. Then, there exists v such that $v^a = x, v^b = y$, and $f(v) = a$. Choose $\epsilon \in \mathbb{R}^n_+$ such that $(x + \epsilon) \in \mathcal{U}^a$. Since f satisfies WPA (Step 1), at profile v' with $v'^a = x + \epsilon$ and $v'^c = v^c$ for all $c \in A \setminus \{a\}$, we have $f(v') = a$ (note that such v' exists due to richness of type space). Hence, $(a, x + \epsilon) \rhd^f (b, y)$.

Step 4. Finally, this implies that $\rhd^f$ is transitive. Suppose $a, b, c \in \bar{A}$ are three distinct alternatives and pick $(a, x), (b, y), (c, z) \in \bar{\mathcal{X}}$ such that $(a, x) \rhd^f (b, y) \rhd^f (c, z)$. Since $(a, x) \rhd^f (b, y)$, there exists a type profile v such that $v^a = x, v^b = y$, and $f(v) = a$. Note that this implies that $(a, x) \rhd^f (a', v^{a'})$ for all $a' \in \bar{A} \setminus \{a\}$. Consider a payoff profile v', where $v'^c = z$ and $v'^{a'} = v^{a'}$ for all $a' \in A \setminus \{c\}$. Since $(a, x) \rhd^f (a', v'^{a'})$ for all $a' \in \bar{A} \setminus \{a, c\}$, $f(v') \in \{a, c\}$. If $f(v') = c$, then $(c, z) \rhd^f (b, y)$, which is a contradiction, since $\rhd^f$ is anti-symmetric (Step 2). Hence, $f(v') = a$, which implies that $(a, x) \rhd^f (c, z)$.

The other case is $(a, x + \epsilon) \rhd^f (a, x) \rhd^f (b, y)$ for some $\epsilon \in \mathbb{R}^n_+$ with $x \neq (x + \epsilon)$ and $x, (x + \epsilon) \in \mathcal{U}^a$. But by Step 3, $(a, x + \epsilon) \rhd^f (b, y)$.

Finally, the case $(b, y) \rhd^f (a, x+\epsilon) \rhd^f (a, x)$, where $\epsilon \in \mathbb{R}^n_+$ and $x \neq (x+\epsilon)$, $x, (x+\epsilon) \in \mathcal{U}^a$. Since $(b, y) \rhd^f (a, x + \epsilon)$, there exists a profile v with $v^b = y, v^a = x + \epsilon$, and $f(v) = b$. This implies that $(b, y) \rhd^f (a', v^{a'})$ for all $a' \in \bar{A} \setminus \{b\}$. Now, consider the profile v' where $v'^a = x, v'^{a'} = v^{a'}$ for all $a' \neq a$ (by richness, such a type profile exists). By binary independence, $f(v') \in \{b, a\}$. If $f(v') = a$, then $(a, x) \rhd^f (b, y)$ and Step 3 implies that

$(a, x+\epsilon) \rhd^f (b, y)$, which is a contradiction. Hence, $f(v') = b$, and this implies that $(b, y) \rhd^f (a, x)$.

STEP 5. This shows that $\rhd^f$ is an irreflexive, anti-symmetric, transitive binary relation on $\bar{\mathcal{X}}$. By Szpilrajn's extension theorem, we can extend it to a complete, irreflexive, anti-symmetric, transitive binary relation on $\bar{\mathcal{X}}$. Since $\bar{\mathcal{X}}$ is finite, there is a payoff representation $\bar{W} : \bar{\mathcal{X}} \to \mathbb{R}$ of this linear order. We can then extend this map to $W : \mathcal{X} \to \mathbb{R}$ as follows, for every $(a, x) \in \bar{\mathcal{X}}$, let $W(a, x) := \bar{W}(a, x)$. Then choose $\delta < \min_{(a,x)\in\bar{\mathcal{X}}} \bar{W}(a, x)$, and set $W(a, x) := \delta$ for every $(a, x) \notin \bar{\mathcal{X}}$.

Now, since $\rhd^f$ satisfies $(a, x+\epsilon) \rhd^f (a, x)$ for all $a \in \bar{A}$, for all $x, (x+\epsilon) \in \mathcal{U}^a$ with $\epsilon \in \mathbb{R}^n$ and $x \neq (x+\epsilon)$, W is monotone. Now, at every profile v, if $f(v) = a$, by definition, $(a, v^a) \rhd^f (b, v^b)$ for all $b \in \bar{A} \setminus \{a\}$, which implies that $W(a, v^a) > W(b, v^b)$ for all $b \neq a$. Hence, W is an APM. Further, note that $\bar{W}$ is an injective map. Hence, no tie-breaking is necessary for W. So, vacuously, it is an APM with consistent tie-breaking.

REFERENCES

ABHISHEK, V., B. HAJEK, AND S. R. WILLIAMS (2013): "Auctions with a Profit Sharing Contract," *Games and Economic Behavior*, 77, 247–270.

AFRIAT, S. N. (1967): "The Construction of Utility Functions from Expenditure Data," *International Economic Review*, 8, 67–77.

BIKHCHANDANI, S., S. CHATTERJI, R. LAVI, A. MUALEM, N. NISAN, AND A. SEN (2006): "Weak Monotonicity Characterizes Deterministic Dominant Strategy Implementation," *Econometrica*, 74, 1109–1132.

BROWN, D. J. AND C. CALSAMIGLIA (2007): "The Nonparametric Approach to Applied Welfare Analysis," *Economic Theory*, 31, 183–188.

CHE, Y.-K. AND J. KIM (2010): "Bidding with Securities: Comment," *American Economic Review*, 100, 1929–1935.

CRÉMER, J. (1987): "Auctions with contingent payments: Comment," *American Economic Review*, 77, 746.

D'ASPREMONT, C. AND L. GEVERS (2002): *Handbook of Social Choice and Welfare - Volume 2*, Elsevier, Amsterdam, chap. Social Welfare Functionals and Interpersonal Comparability (Chapter 10), 459–541, ediors: Kenneth J. Arrow, Amartya K. Sen, and K. Suzumura.

DEB, R. AND D. MISHRA (2014): "Supplement to 'Implementation with Contingent Contracts'," Available online.

DeMarzo, P. M., I. Kremer, and A. Skrzypacz (2005): "Bidding with Securities: Auctions and Security Design," *American Economic Review*, 95, 936–959.

Ekmekci, M., N. Kos, and R. Vohra (2014): "Just Enough or All: Selling a Firm," Working Paper.

Green, J. R. and J.-J. Laffont (1979): *Incentives in Public Decision Making*, North-Holland.

Hansen, R. G. (1985): "Auctions with Contingent Payments," *American Economic Review*, 75, 862–865.

Jehiel, P. and B. Moldovanu (2001): "Efficient Design with Interdependent Valuations," *Econometrica*, 69, 1237–1259.

Kos, N. and M. Messner (2013): "Extremal Incentive Compatible Transfers," *Journal of Economic Theory*, 148, 134–164.

Laffont, J.-J. and J. Tirole (1986): "Using Cost Observation to Regulate Firms," *Journal of Political Economy*, 614–641.

Mezzetti, C. (2004): "Mechanism Design with Interdependent Valuations: Efficiency," *Econometrica*, 72, 1617–1626.

Milgrom, P. R. and R. J. Weber (1982): "A Theory of Auctions and Competitive Bidding," *Econometrica*, 50, 1089–1122.

Rahman, D. (2011): "Detecting Profitable Deviations," Working Paper, University of Minnesota.

Roberts, K. (1979): *The Characterization of Implementable Choice Rules*, North Holland Publishing, chap. Aggregation and Revelation of Preferences, 321–348, editor: J-J. Laffont.

Rochet, J. C. (1987): "A Necessary and Sufficient Condition for Rationalizability in a Quasi-linear Context," *Journal of Mathematical Economics*, 16, 191–200.

Rockafellar, R. T. (1970): *Convex Analysis*, Princeton University Press.

Sen, A. K. (1970): *Collective Choice and Social Welfare*, vol. 5, Holden-Day.

Skrzypacz, A. (2013): "Auctions with Contingent Payments - An Overview," *International Journal of Industrial Organization*.

Varian, H. R. (1982): "The Nonparametric Approach to Demand Analysis," *Econometrica*, 50, 945–973.

Vohra, R. V. (2011): *Mechanism Design: A Linear Programming Approach*, Cambridge University Press.

Walker, M. (1980): "On the nonexistence of a dominant strategy mechanism for making optimal public decisions," *Econometrica*, 1521–1540.