

# ECONOMETRICA

JOURNAL OF THE ECONOMETRIC SOCIETY

*An International Society for the Advancement of Economic  
Theory in its Relation to Statistics and Mathematics*

<http://www.econometricsociety.org/>

*Econometrica*, Vol. 82, No. 6 (November, 2014), 2371–2393

## IMPLEMENTATION WITH CONTINGENT CONTRACTS

RAHUL DEB

*University of Toronto, Toronto, ON M5S 3G7, Canada*

DEBASIS MISHRA

*Indian Statistical Institute, Delhi, New Delhi 110016, India*

---

The copyright to this Article is held by the Econometric Society. It may be downloaded, printed and reproduced only for educational or research purposes, including use in course packs. No downloading or copying may be done for any commercial purpose without the explicit permission of the Econometric Society. For such commercial purposes contact the Office of the Econometric Society (contact information may be found at the website <http://www.econometricsociety.org> or in the back cover of *Econometrica*). This statement must be included on all copies of this Article that are made available electronically or in any other format.

---

## IMPLEMENTATION WITH CONTINGENT CONTRACTS

BY RAHUL DEB AND DEBASIS MISHRA<sup>1</sup>

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.

**KEYWORDS:** Implementation, dominant strategies, contingent contracts, securities, linear contracts, social welfare.

### 1. INTRODUCTION

THE CLASSIC SETTING IN MECHANISM DESIGN with quasilinear 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 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) that 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.

<sup>1</sup>This paper is a significant revision of an earlier paper titled "Implementation With Securities." We are grateful to a co-editor 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.

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, and wireless spectrum and highway building contracts.<sup>2</sup>

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) that 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.

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 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.

<sup>2</sup>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.

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 that 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 (VCG) transfers (Green and Laffont (1979)).

### 1.1. *Related 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 that discusses the revenue ranking of auctions conducted with different contingent contracts (DeMarzo, Kremer, and Skrzypacz (2005), Che and Kim (2010), Abhishek, Hajek, and Williams (2013)). These papers study how a seller's revenue is affected by the “steepness” of securities that are admissible as bids.

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 the real world for a number of reasons. For instance, the principal may be liquidity constrained and, hence, may be unable to finance the necessary up-front payment to buy the agent. In other cases, as DeMarzo, Kremer, and Skrzypacz (2005) argue, the agents may have to make noncontractible, fixed, costly investments 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 up-front 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, Kos, and Vohra (2014)). In practice, environments that feature contingent contracts often have various legal or practical restrictions that may prevent the principle from extracting all the surplus. For these applications, our characterization of incentive compatibility is an important first step that 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 that 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 of both the signal and the agent's re-

port. 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  $v_i: A \rightarrow \mathbb{R}$  and  $V_i$  denotes the set of all possible types of agent  $i$ . Using the standard notation,  $V := V_1 \times \dots \times V_n$  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. 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 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  $f: V \rightarrow A$ . 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 that has this feature is a *linear contract*. A linear contract for agent  $i$  consists of two mappings: a royalty (or tax) rule  $r_i: V \rightarrow (0, 1]$  and a transfer rule  $t_i: V \rightarrow \mathbb{R}$ . A *linear mechanism*  $(f, (r_1, t_1), \dots, (r_n, t_n))$  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 among reports and trivializing the implementation problem.

A special case of the linear mechanism is the standard *quasilinear mechanism*  $(f, t_1, \dots, t_n)$ , 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 quasilinear 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 that satisfy this property. A *contingent contract* of agent  $i$  is a map  $s_i: \mathbb{R} \times V \rightarrow \mathbb{R}$  that 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  $(f, s_1, \dots, s_n)$ , where  $f$  is an SCF and  $(s_1, \dots, s_n)$  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 that are weakly increasing such as call options and convertible debt. Again, this assumption is made to prevent the principal from making agents indifferent among 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 that 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: 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  $\forall i \in N$ ,  $\forall v_{-i} \in V_{-i}$ ,

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

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  $\forall i \in N, \forall v_{-i} \in V_{-i}$ ,

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

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, Chatterji, Lavi, Mualem, Nisan, and Sen (2006) (see their Supplemental Material) of an SCF that 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  $A = \{a^1, a^2\}$ . The payoffs are given as

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

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

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

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 at least 2 in transfers and a loss of only 1 from payoff from the worse alternative  $a^1$ .

Instead, consider the linear contract for agent 1:

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

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 the 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 is required for implementation.

### 3.1. A Foundation for Linear Contracts

We now characterize the set of SCFs that are implementable by linear contracts and in doing so we will additionally 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  $\succeq_{v_{-i}}^f$  and  $\succ_{v_{-i}}^f$  on  $V_i$  as follows. For notational convenience, we write  $\succeq^f \equiv \succeq_{v_{-i}}^f$  and  $\succ^f \equiv \succ_{v_{-i}}^f$ ; 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, or 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 be neither 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  $v'_i \not\succeq^f v_i$ .

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

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.<sup>3</sup>

LEMMA 1: *If an SCF is implementable, it is acyclic.*

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

$$\begin{aligned} s_i(v_i^j(f(v_i^j, v_{-i})), v_i^j, v_{-i}) &\geq s_i(v_i^j(f(v_i^{j+1}, v_{-i})), v_i^{j+1}, v_{-i}) \\ &\geq s_i(v_i^{j+1}(f(v_i^{j+1}, v_{-i})), v_i^{j+1}, v_{-i}), \end{aligned}$$

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  $j \in \{1, \dots, k-1\}$ , we get

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

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,  $v_i^k \not\prec v_i^1$ , which implies that  $f$  is acyclic. *Q.E.D.*

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 given in the [Appendix](#).

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 1: For every acyclic SCF, the proof explicitly constructs a linear contract that implements it. Under an additional 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 nonnegative (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.<sup>4</sup>

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

<sup>4</sup>This is true even for standard quasilinear mechanisms: every implementable SCF can be implemented (under reasonable conditions) using individually rational and nonnegative transfers ([Kos and Messner \(2013\)](#)).

REMARK 2: Theorem 1 uncovers a parallel with Afriat's theorem (Afriat (1967)) of revealed preference in consumer theory.<sup>5</sup> 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. Furthermore, 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.<sup>6</sup>

It is important to point out that the finite type space assumption is required for the equivalence between implementability by contingent and linear contracts. In infinite type spaces, this equivalence may break down due to discontinuities either in the SCF or in the type to payoff mapping for a given alternative (a counterexample with a countably infinite type space is given in the Supplemental Material (Deb and Mishra (2014))). However, Theorem B in the Supplemental Material shows that the equivalence continues to hold in uncountable type spaces under additional smoothness conditions.

### 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  $v \equiv (v_1, \dots, v_n) \in \mathcal{V}$ , we can define a vector  $v^a \equiv (v_1(a), \dots, v_n(a)) \in \mathbb{R}^n$  for each alternative  $a \in A$ . This is the payoff vector of the agents corresponding to alternative  $a$ . Given a type space  $\mathcal{V}$ , 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  $\mathcal{U}^a$ .

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

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

An *aggregate payoff function* is a map  $W : \mathcal{X} \rightarrow \mathbb{R}$ . 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  $W(a, y) \geq W(a, x)$ .<sup>7</sup>

<sup>5</sup>Beginning with Rochet (1987), there have been informal analogies made between these two problems.

<sup>6</sup>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)).

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

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

$$f(v) \in \arg \max_{a \in A} W(a, v^a).$$

Furthermore, 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 nonnegative weights  $(\gamma_1, \dots, \gamma_n) \in \mathbb{R}_+^n \setminus \{0\}$  and a map  $\kappa : A \rightarrow \mathbb{R}$  such that for all  $v \in V$ ,

$$f(v) \in \arg \max_{a \in A} \left[ \sum_{i \in N} \gamma_i v_i(a) - \kappa(a) \right].$$

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

$$f(v) \in \arg \max_{a \in A} \min_{i \in N} v_i(a).$$

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  $f$ . Furthermore, let  $W$  be a monotone aggregate payoff function such that  $f(v) \in \arg \max_{a \in A} W(a, v^a)$  for every  $v \in V$ .

Fix an agent  $i \in N$  and type profile of other agents at  $v_{-i}$ . Consider a sequence of types  $v_i^1 \succeq^f \dots \succeq^f v_i^k$ . Pick any  $j \in \{1, \dots, k - 1\}$ . 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  $j \in \{1, \dots, k - 1\}$ , we get that

$$\begin{aligned} W(a_1, v^{1,a_1}) &\geq W(a_2, v^{1,a_2}) \geq W(a_2, v^{2,a_2}) \geq \dots \geq \dots \\ &\geq W(a_k, v^{k-1,a_k}) \geq W(a_k, v^{k,a_k}). \end{aligned}$$

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

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

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

$$\begin{aligned} W(a_1, v^{1,a_1}) &= W(a_2, v^{1,a_2}) = W(a_2, v^{2,a_2}) = \dots \\ &= W(a_k, v^{k,a_k}) = W(a_1, v^{k,a_1}) = W(a_1, v^{1,a_1}). \end{aligned}$$

Now, pick any  $j \in \{1, \dots, k - 1\}$ . 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  $j \in \{1, \dots, k - 1\}$ , we see that either  $a_1 = a_2 = \dots = a_k$  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,  $a_1 = a_2 = \dots = a_k = a$  for some  $a \in A$ . But this implies that  $v_i^1(a) \geq v_i^2(a) \geq \dots \geq v_i^k(a)$ , and this contradicts that  $v_i^k \succ^f v_i^1$ . Q.E.D.

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  $\mathcal{U}^a \times \mathcal{U}^b \times \dots$ .

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  $\mathcal{U} := \mathcal{U}^a \times \mathcal{U}^b \times \dots$ .

We now impose an independence condition on the SCFs. This is in the spirit of binary independence used in the social choice theory literature (d'Aspremont and Gevers (2002)) from which 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  $v^a = v'^a$  and  $v^b = v'^b$ ,  $f(v) = a$  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 independently 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 given in the [Appendix](#).

**THEOREM 3:** *In a finite and rich type space, the following statements are equivalent for an SCF  $f$ :*

- (i)  $f$  is an aggregate payoff maximizer with consistent tie-breaking.
- (ii)  $f$  is implementable and satisfies binary independence.
- (iii)  $f$  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 quasilinear mechanisms. Roberts (1979) showed that affine maximizers are the only implementable SCFs using quasilinear mechanisms. Though Theorem 3 can be viewed as a 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  $(s_1, \dots, s_n)$  that implement it, we ask if there exist linear contracts  $((r_1, t_1), \dots, (r_n, t_n))$  that implement  $f$  such that

$$s_i(v_i(f(v_i, v_{-i})), v_i, v_{-i}) = r_i(v_i, v_{-i})v_i(f(v_i, v_{-i})) - t_i(v_i, v_{-i})$$

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.<sup>8</sup>

<sup>8</sup>The failure of payoff equivalence is not driven by the finite type space restriction and it is easy to construct similar examples with a continuum of types. Furthermore, the usual revenue/payoff

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

	$v_1^1$	$v_1^2$	$v_1^3$
$a^1$	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 contingent contract

$$\begin{aligned}
 s_1(30, v_1^1) &= 20, & s_1(20, v_1^1) &= 5, & s_1(10, v_1^1) &= 1, \\
 s_1(30, v_1^2) &= 16, & s_1(20, v_1^2) &= 15, & s_1(10, v_1^2) &= 1, \\
 s_1(20, v_1^3) &= 10, & s_1(10, v_1^3) &= 5.
 \end{aligned}$$

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

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

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

$$(2) \quad s_1(v_1'(f(v_1')), v_1') - s_1(v_1(f(v_1)), v_1) \leq r_1(v_1') [v_1'(f(v_1')) - v_1(f(v_1))].$$

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

Though payoff equivalence does not hold between linear mechanisms and nonlinear 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  $f^*(v) \in \arg \max_{a \in A} \sum_{i \in N} v_i(a)$ , 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 simple linear contract

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

---

equivalence in quasilinear 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  $G_{v_i}^a$ , 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.<sup>9</sup> In a minor abuse of notation, we use  $v_i(a)$  to denote the expected payoff from alternative  $a$  or  $v_i(a) = \int_{\mathbb{R}} u_i dG_{v_i}^a(u_i)$ .

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

**DEFINITION 7:** The distributions of payoffs are ordered by first-order stochastic dominance (FOSD) or simply *ordered* if for all  $i, v_i, v'_i \in V_i$  and for all  $a \in A$ , we have

$$\text{either } G_{v_i}^a \succeq_{\text{FOSD}} G_{v'_i}^a \quad \text{or} \quad G_{v'_i}^a \succeq_{\text{FOSD}} G_{v_i}^a,$$

where  $\succeq_{\text{FOSD}}$  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  $\succeq_{\text{FOSD}}$  relation. Note that this does not imply that the ordering of types has to be the same across

<sup>9</sup>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.

the different alternatives. To the best of our knowledge, most of the theoretical work on mechanism design with contingent contracts requires this assumption.<sup>10</sup> 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  $f: V \rightarrow A$ . Linear contracts are defined identically and consist of functions  $r_i: V \rightarrow (0, 1]$  and  $t_i: V \rightarrow \mathbb{R}$  for each agent  $i$ . Similarly, contingent contracts are mappings  $s_i: \mathbb{R} \times V \rightarrow \mathbb{R}$  for each agent  $i$  that 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  $(s_1, \dots, s_n)$  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  $(s_1, \dots, s_n)$  implement  $f$  and the contingent mechanism  $(f, s_1, \dots, s_n)$  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  $G_{v_i}^a \succeq_{\text{FOSD}} (>_{\text{FOSD}}) G_{v'_i}^a$ . Finally, observe that if the type space is ordered, acyclicity remains necessary for implementation. This is because for ordered types  $v'_i, v_i \in V_i$ ,

$$\begin{aligned} v'_i \succeq^f v_i &\implies \int_{\mathbb{R}} s_i(u_i, v_i, v_{-i}) dG_{v'_i}^{f(v_i, v_{-i})}(u_i) \\ &\geq \int_{\mathbb{R}} s_i(u_i, v_i, v_{-i}) dG_{v_i}^{f(v_i, v_{-i})}(u_i) \end{aligned}$$

<sup>10</sup>Often, the stronger assumption of affiliation (Milgrom and Weber (1982)) is made instead (Skrzypacz (2013)).



holds, 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.

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 the 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. The results that we describe below can be found in the Supplemental Material (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 co-monotone, the result will continue to hold. Of course, an important extension for future work is to examine environments in which these are not co-monotone where Theorem 1 does not hold in general.

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.

## APPENDIX

### A.1. 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  $(V_i^1, \dots, V_i^K)$  of the type space  $V_i$  such that the following statements hold.

P1. For each  $j \in \{1, \dots, K\}$  and for each  $v_i, v'_i \in V_i^j$ , we have  $v_i \not\prec^f v'_i$ ,

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

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 nonempty 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  $\tilde{V}_i'$ . Since  $f$  is acyclic,  $\succ^f$  is acyclic. Since  $V_i'$  is finite, we conclude that  $\tilde{V}_i'$  is nonempty (Sen (1970)). Define

$$M(V_i') := \{v_i \in \tilde{V}_i' : v_i \not\geq^f v_i' \forall v_i' \in V_i' \setminus \tilde{V}_i'\}.$$

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

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

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  $\lambda_i : V_i \rightarrow (0, \infty)$  such that for all sequence of types  $(v_i^1, \dots, v_i^k)$  with  $k \leq K$ , we have

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

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

To show that  $f$  is s.c.m., we construct a  $\lambda_i$  that makes it s.c.m.

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

$$(4) \quad \lambda_i(v_i) = 1 \quad \forall v_i \in V_i^K.$$

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

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

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

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

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  $f$ -order-partitioned, for every  $v_i \in V_i^k$  and  $v_i' \in (V_i^{k+1} \cup \dots \cup V_i^K)$ , we have  $v_i' \not\leq^f v_i$  (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  $v_i' \not\leq^f v_i$  (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  $(V_i^{k+1} \cup \dots \cup V_i^K)$ . Now, for every  $v_i \in V_i^k$ , define

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

We can thus recursively define the  $\lambda_i$  map.

**PROPOSITION 1:** *Suppose  $V_i$  is finite. If an SCF  $f$  is acyclic, then  $\lambda_i$  makes  $f$  s.c.m.*

<sup>11</sup>We will abuse notation to denote the set of types in a cycle  $C$  by  $C$  also.

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

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

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  $C \subseteq (V_i^{k+1} \cup \dots \cup V_i^K)$ . Consider a cycle  $C \equiv (v_i^1, \dots, v_i^q, v_i^{q+1} \equiv v_i^1)$  involving types in  $(V_i^k \cup \dots \cup V_i^K)$ . 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  $(V_i^{k+1} \cup \dots \cup V_i^K)$ , 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  $(V_i^{k+1} \cup \dots \cup V_i^K)$ . Let  $\lambda_i(v_i) = \mu$  for all  $v_i \in V_i^k$ . By definition,

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

where the last inequality follows from the definition of  $\mu$  (Equation (7)). Hence, inequality (8) again holds. Proceeding in this manner inductively, we complete the proof. Q.E.D.

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

Constructing  $t_i$ : If  $\lambda_i$  makes  $f$  s.c.m., then  $\lambda_i$  satisfies inequality (3) for any cycle of types. Hence, by Rochet–Rockafellar cycle monotonicity characterization (Rochet (1987), Rockafellar (1970)), there exists a map  $W_i: V_i \rightarrow \mathbb{R}$  such that

$$(9) \quad W_i(v_i) - W_i(v_i') \leq \lambda_i(v_i)[v_i(f(v_i)) - v_i'(f(v_i))] \quad \forall v_i, v_i' \in V_i.$$

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

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

PROPOSITION 2: *If  $\lambda_i$  makes  $f$  s.c.m., then  $(\lambda_i, t_i)$  is an incentive compatible linear mechanism.*

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

$$\begin{aligned} &\lambda_i(v_i)v_i(f(v_i)) - t_i(v_i) - \lambda_i(v'_i)v'_i(f(v'_i)) + t_i(v'_i) \\ &\leq \lambda_i(v_i)[v_i(f(v_i)) - v'_i(f(v_i))]. \end{aligned}$$

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

$$\lambda_i(v'_i)v'_i(f(v'_i)) - t_i(v'_i) \geq \lambda_i(v_i)v'_i(f(v_i)) - t_i(v_i). \tag{Q.E.D.}$$

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  $a \in A$ . Furthermore, 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 nonnegative. 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

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

Similarly, for any  $v_i$ ,

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

Hence,  $t_i(v_i) \geq 0$ . Finally, note that we can always scale  $(\lambda_i, t_i)$  such that  $\lambda_i$  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 nonnegative and all agents are individually rational.

### A.2. Proof of Theorem 3

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

(iii)  $\Rightarrow$  (i). 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  $\bar{A} := \{a \in A : \text{there exists } v \in V \text{ such that } f(v) = a\}$ , that is,  $\bar{A}$  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  $\triangleright^f$  on the elements of  $\bar{\mathcal{X}}$ . For any  $(a, x), (b, y) \in \bar{\mathcal{X}}$  with  $a \neq b$ , we let

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

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

$$(a, x + \varepsilon) \triangleright^f (a, x).$$

Note that the binary relation is only a partial order. Binary independence immediately implies that  $\triangleright^f$  is antisymmetric.

STEP 3: We will say that the binary relation  $\triangleright^f$  satisfies a monotonicity property. Pick distinct  $a, b \in \bar{A}$  and  $x \in \mathcal{U}^a, y \in \mathcal{U}^b$  such that  $(a, x) \triangleright^f (b, y)$ . Then there exists  $v$  such that  $v^a = x, v^b = y$ , and  $f(v) = a$ . Choose  $\varepsilon \in \mathbb{R}_+^n$  such that  $(x + \varepsilon) \in \mathcal{U}^a$ . Since  $f$  satisfies WPA (Step 1), at profile  $v'$  with  $v'^a = x + \varepsilon$  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,  $(a, x + \varepsilon) \triangleright^f (b, y)$ .

STEP 4: Finally, this implies that  $\triangleright^f$  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  $(a, x) \triangleright^f (b, y) \triangleright^f (c, z)$ . Since  $(a, x) \triangleright^f (b, y)$ , there exists a type profile  $v$  such that  $v^a = x, v^b = y$ , and  $f(v) = a$ . Note that this implies that  $(a, x) \triangleright^f (a', v^{a'})$  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  $(a, x) \triangleright^f (a', v^{a'})$  for all  $a' \in \bar{A} \setminus \{a, c\}$ ,  $f(v') \in \{a, c\}$ . If  $f(v') = c$ , then  $(c, z) \triangleright^f (b, y)$ , which is a contradiction, since  $\triangleright^f$  is antisymmetric (Step 2). Hence,  $f(v') = a$ , which implies that  $(a, x) \triangleright^f (c, z)$ .

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

Finally, the case  $(b, y) \triangleright^f (a, x + \varepsilon) \triangleright^f (a, x)$ , where  $\varepsilon \in \mathbb{R}_+^n$  and  $x \neq (x + \varepsilon)$ ,  $x, (x + \varepsilon) \in \mathcal{U}^a$ . Since  $(b, y) \triangleright^f (a, x + \varepsilon)$ , there exists a profile  $v$  with  $v^b = y, v^a = x + \varepsilon$ , and  $f(v) = b$ . This implies that  $(b, y) \triangleright^f (a', v^{a'})$  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  $(a, x) \triangleright^f (b, y)$  and Step 3 implies that  $(a, x + \varepsilon) \triangleright^f (b, y)$ , which is a contradiction. Hence,  $f(v) = b$ , and this implies that  $(b, y) \triangleright^f (a, x)$ .

STEP 5: This shows that  $\triangleright^f$  is an irreflexive, antisymmetric, transitive binary relation on  $\bar{\mathcal{X}}$ . By Szpilrajn's extension theorem, we can extend it to a complete, irreflexive, antisymmetric, transitive binary relation on  $\bar{\mathcal{X}}$ . Since  $\bar{\mathcal{X}}$  is finite, there is a payoff representation  $\bar{W} : \bar{\mathcal{X}} \rightarrow \mathbb{R}$  of this linear order. We can then extend this map to  $W : \mathcal{X} \rightarrow \mathbb{R}$  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  $\triangleright^f$  satisfies  $(a, x + \varepsilon) \triangleright^f (a, x)$  for all  $a \in \bar{A}$ , for all  $x, (x + \varepsilon) \in \mathcal{U}^a$  with  $\varepsilon \in \mathbb{R}^n$ , and  $x \neq (x + \varepsilon)$ ,  $W$  is monotone. Now, at every profile  $v$ , if  $f(v) = a$ , by definition,  $(a, v^a) \triangleright^f (b, v^b)$  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. Furthermore, 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. [2373]
- AFRIAT, S. N. (1967): "The Construction of Utility Functions From Expenditure Data," *International Economic Review*, 8, 67–77. [2379]
- 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. [2376]
- BROWN, D. J., AND C. CALSAMIGLIA (2007): "The Nonparametric Approach to Applied Welfare Analysis," *Economic Theory*, 31 (1), 183–188. [2379]
- CHE, Y.-K., AND J. KIM (2010): "Bidding With Securities: Comment," *American Economic Review*, 100, 1929–1935. [2373]
- CRÉMER, J. (1987): "Auctions With Contingent Payments: Comment," *American Economic Review*, 77 (4), 746. [2373]
- D'ASPROMONT, C., AND L. GEVERS (2002): "Social Welfare Functionals and Interpersonal Comparability," in *Handbook of Social Choice and Welfare*, Vol. 2, ed. by K. J. Arrow, A. K. Sen, and K. Suzumura. Amsterdam: Elsevier, 459–541, Chapter 10. [2381,2382]
- DEB, R., AND D. MISHRA (2014): "Supplement to 'Implementation With Contingent Contracts'," *Econometrica Supplemental Material*, 82, <http://dx.doi.org/10.3982/ECTA11561>. [2379,2386]
- DEMARZO, P. M., I. KREMER, AND A. SKRZYPACZ (2005): "Bidding With Securities: Auctions and Security Design," *American Economic Review*, 95, 936–959. [2373]
- EKMEKCI, M., N. KOS, AND R. VOHRA (2014): "Just Enough or All: Selling a Firm," Working Paper. [2373]
- GREEN, J. R., AND J.-J. LAFFONT (1979): *Incentives in Public Decision Making*. Amsterdam: North-Holland. [2373]
- HANSEN, R. G. (1985): "Auctions With Contingent Payments," *American Economic Review*, 75, 862–865. [2373]
- KOS, N., AND M. MESSNER (2013): "Extremal Incentive Compatible Transfers," *Journal of Economic Theory*, 148, 134–164. [2378]

- MILGROM, P. R., AND R. J. WEBER (1982): "A Theory of Auctions and Competitive Bidding," *Econometrica*, 50, 1089–1122. [2385]
- RAHMAN, D. (2011): "Detecting Profitable Deviations," Working Paper, University of Minnesota. [2373]
- ROBERTS, K. (1979): "Aggregation and Revelation of Preferences," in *The Characterization of Implementable Choice Rules*, ed. by J.-J. Laffont. Amsterdam: North-Holland, 321–348. [2382]
- ROCHET, J. C. (1987): "A Necessary and Sufficient Condition for Rationalizability in a Quasi-Linear Context," *Journal of Mathematical Economics*, 16, 191–200. [2373,2378,2379,2389]
- ROCKAFELLAR, R. T. (1970): *Convex Analysis*. Princeton: Princeton University Press. [2389]
- SEN, A. K. (1970): *Collective Choice and Social Welfare*, Vol. 5. San Francisco: Holden-Day. [2387]
- SKRZYPACZ, A. (2013): "Auctions With Contingent Payments—An Overview," *International Journal of Industrial Organization*, 31 (5), 666–675. [2373,2385]
- VARIAN, H. R. (1982): "The Nonparametric Approach to Demand Analysis," *Econometrica*, 50, 945–973. [2379]
- VOHRA, R. V. (2011): *Mechanism Design: A Linear Programming Approach*. New York: Cambridge University Press. [2389]
- WALKER, M. (1980): "On the Nonexistence of a Dominant Strategy Mechanism for Making Optimal Public Decisions," *Econometrica*, 48, 1521–1540. [2383]

*University of Toronto, 150 St. George St., Toronto, ON M5S 3G7, Canada;*  
*rahul.deb@utoronto.ca*

*and*

*Indian Statistical Institute, Delhi, New Delhi 110016, India; dmishra@*  
*isid.ac.in.*

*Manuscript received April, 2013; final revision received July, 2014.*