Implementation by Agenda Voting

Sean Horan

Abstract

Agenda voting occurs in a wide variety of contexts. This paper characterizes the class of social choice functions that can be implemented by sophisticated voting on an agenda under the assumption of complete information. The main result establishes that a simple *pairwise condition* is necessary and sufficient for implementation by agenda voting.
Keywords: Sophisticated voting, implementation, voting agendas.

1 Introduction

Voting by agenda occurs in a wide variety of political and social choice contexts. The economic analysis of agendas has a rich tradition in the literature dating back to the early work of Black [1958] and Farquharson [1969]. This paper contributes to that literature by characterizing the social choice functions that can be implemented by *sophisticated* voting in an environment with complete information. The main result establishes that a simple *pairwise condition* defined on pairs of states is necessary and sufficient for implementation. The paper builds on earlier work by Srivastava and Trick [1996], who conjectured that a weaker condition defined on pairs of *adjacent* states (i.e. states that differ on the ranking of two outcomes) was necessary and sufficient.[1]

Formally, a voting agenda describes a binary tree where, at any decision node, the agents vote between two collections of competing proposals. Ultimately, the winning proposal is the outcome that survives the sequence of binary votes given by the agenda. If the agents are forward-looking, their behavior is *sophisticated* and the winning proposal can be determined by backward induction. Provided that the provisional winners are determined by simple majority, the winning proposal must be drawn from the *Condorcet set* — the subset of outcomes that indirectly dominates every outcome (see Miller [1977], McKelvey-Niemi [1978], Moulin [1986]). Because the *pairwise condition* is relatively weak, agenda voting is capable of implementing a wide variety of selections from the Condorcet set.

A variety of approaches can be used to implement outcomes from the Condorcet set. Most closely related to implementation by agenda voting are the extensive-form mechanisms based on *backward induction* (Gol'berg-Gourvitch [1986] and Herrero-Srivastava [1992]) and *subgame perfection* (Abreu-Sen [1990], Moore-Repullo [1988], and Vartiainen [2007a]). Also related to implementation by agenda voting are the normal-form solution concepts based on *dominance solvable voting* (Moulin [1979]) and *undominated Nash equilibrium* (Palfrey-Srivastava [1989]).

While each of these four mechanisms is capable of implementing a wider variety of outcomes than agenda voting (see e.g. Dutta-Sen [1993]), there are some compelling advantages to the approach taken here. Perhaps most importantly, agenda voting is a straightforward way to decentralize choice. In contrast with many other approaches to implementation, agenda voting *expressly* rules out artificial features like randomization (see e.g. Vartiainen [2007b]), nuisance strategies (e.g. integer games and *bad* outcomes), and unnaturally complex strategy sets.[2] Arguably, the simplicity of agenda voting is a large part of the reason that this mechanism is so widely used in real-world settings.

No less attractive is the fact that the necessary and sufficient condition for implementation

[1] Their conjecture replaces an earlier conjecture due to Herrero and Srivastava [1992].

[2] However, the agenda required to implement the desired outcomes may be large (see Trick [2006, 2009]).

by agenda voting is simple. By comparison, both of the extensive-form mechanisms discussed above impose necessary conditions that can be quite difficult to verify in practice. In view of this shortcoming, Moore [1992] has stressed the importance of finding a "full and workable characterization of social choice functions that can be implemented in trees." The main result goes some way towards achieving this goal. As discussed, it provides a workable characterization for a fairly broad class of social choice functions that can be implemented in trees. Moreover, it also provides a simple sufficient condition for implementation in trees more generally. This follows from the fact that agenda voting is a special case of implementation via backward induction.

Before moving on, it is worth noting that the approach taken in this paper is somewhat unconventional from a technical standpoint. In the implementation literature, sufficiency of the characterization is generally established by constructing a mechanism that implements any social choice function with the prescribed features. Unusually, the sufficiency proof given here is obtained by algebraic methods that do not rely on the explicit construction of a mechanism. The basic idea of the proof is that extensive-form games can be "added" together at the root to form a new game. The strength of this approach is that the equilibrium of the new game is easily determined from the equilibria of the original games. Since the intuition is straightforward, it is perhaps surprising that very little work in implementation theory leverages the algebraic structure of extensive-form games. The only notable exception is the characterization of implementation via backward induction given by Gol'berg-Gourvitch [1986].

2 Implementation by Agenda Voting

Before stating the main result, this section provides some preliminary definitions and gives some examples of voting agendas that are widely discussed in the literature of social choice and political economy. A discussion of the result is given in Section 3.

2.1 Definitions

Let X denote some finite set of outcomes. The population of agents is given by $A = \{1, ..., a\}$ where $a = |A|$ is odd. Let $\mathcal{L}$ denote the collection of linear orders on X. An element $P=(\succ_1, ..., \succ_a)$ of $\mathcal{L}^a$ represents a profile of individual preference orders on X. For any profile $P \in \mathcal{L}^a$, the *majority relation* R is defined by xRy iff $|\{i \in A : x \succ_i y\}| > |\{i \in A : y \succ_i x\}|$. Since $|A|$ is odd, any majority relation R is a complete, asymmetric relation (or *pairwise ranking*) on X. Let $\mathcal{R}$ denote the collection of majority relations on X.

A *social choice function* (SCF) is a mapping $F : \mathcal{L}^a \to X$ that selects an outcome for every profile $P \in \mathcal{L}^a$. A *Condorcet social choice function* is an SCF that selects the *same outcome* when the majority relations on P and P' coincide (i.e. $R = R'$). In other words, it can be described as a mapping $f : \mathcal{R} \to X$ that selects an outcome for every majority relation $R \in \mathcal{R}$. In what follows, I frequently abbreviate by referring to a majority relation R as a *state*.

Generically, a voting agenda can be described as a labelled binary tree. A *binary tree* B is a pair $(V, <)$ consisting of a finite set V of *vertices* and a strict (but incomplete) transitive order $<$ on V. The order $<$ has a particular structure so that: every vertex has either zero or two *successors* and all vertices except one have a unique *predecessor*. The $<$-maximal vertices in V, denoted by V_0, are the *leaves* of the tree and the unique $<$-minimal vertex v^* is the *root*. In order to label the leaves V_0 of a binary tree B with the alternatives in X (where $|X| \leq |V_0|$), let $\iota : V_0 \to X$ define a surjection, or *seeding*, from the leaves to the elements of X. Together, the binary tree B and the seeding ι define a *voting agenda* $T = (B, \iota)$ over the alternatives in X.

For any voting agenda T and majority relation R on X, the *overall winner* $c_T(R) = v^*(T, R)$ is determined by backward induction. The winner $v(T, R)$ at any leaf $v \in V_0$ is the alternative

$\iota(v)$ that labels v and the winner at any non-leaf $v \notin V_0$ is given by majority voting between the winners at the left successor v_l of v and right successor v_r. Formally:

$$v(T, R) \equiv \begin{cases} v_l(T; R) & \text{if } (v_l(T; R), v_r(T; R)) \in R \\ \\ v_r(T; R) & \text{otherwise} \end{cases}$$

Definition 1 (Implementation by Agenda Voting) *A Condorcet SCF f is* ***implementable*** *if there exists an agenda T on X such that $c_T(R) = f(R)$ for every state $R \in \mathcal{R}$.*

Before moving on, I pause to make two comments about this definition. First, observe that it requires implementation for *all* possible states (i.e. every majority relation on X). In the literature, this is known as the *universal domain* assumption. Second, it requires that the agenda implementing f contain *every* alternative in X. Given the assumption of universal domain, this is without loss of generality. The reason is that agenda voting must select the *Condorcet winner* – i.e. the item $x \in X$ s.t. xRy for all $y \in X \setminus \{x\}$ – whenever it exists. Because every alternative in X is the Condorcet winner for some state(s), every alternative is chosen in some state — and, hence, must be part of the agenda.

When there is no Condorcet winner, agenda voting must select from the Condorcet set:

Definition 2 (Condorcet Set) *The* ***Condorcet set*** $\mathbf{C}(Y, R)$ *of the pairwise ranking R on Y is the smallest subset of Y where yRy' for all $y \in \mathbf{C}(Y, R)$ and $y' \in A \backslash \mathbf{C}(Y, R)$. When R is understood, I abbreviate to $\mathbf{C}(Y)$.*

In other words, $\mathbf{C}(Y)$ is a (possibly degenerate) cycle in Y whose members pairwise-dominate every outcome in $Y \backslash \mathbf{C}(Y)$ (see e.g. Moulin [1986] and Laslier [1997]). Intuitively, the Condorcet set generalizes the usual notion of maximization to address the situation where no single outcome R-dominates every other outcome in Y.

2.2 Main Result

The main result establishes that f is implementable if it is implementable for all pairs of majority relations. Formally, outcomes x and x' are *pairwise implementable* in states R and R' if there exists a voting agenda T such that $c_T(R) = x$ and $c_T(R') = x'$. To state the main result:

Main Result *A Condorcet SCF f is implementable iff is implementable for every pair of states.*

Based on the work of Srivastava and Trick [1996], it can be shown that any outcomes x and x' in the Condorcet sets of R and R' are pairwise implementable if the two states are sufficiently distinct on a *global* level. If the states are globally similar however, one can only implement outcomes from the same *locale*. Some definitions are required to formalize these notions.

Given a pairwise-ranking R on Y, a subset $Y' \subseteq Y$ is a *component* of R if every element in Y' bears the same relation to elements in $Y \backslash Y'$. Given an item $y \in Y \backslash Y'$ and any items y', $y'' \in Y'$, then $y''Ry$ if and only if $y'Ry$.[3] A *decomposition* of a pairwise-ranking R on $Y \subseteq X$ is a partition of Y into components. The largest decomposition is the *degenerate partition* $\{Y\}$.

If R is *cyclic* on Y (so that $\mathbf{C}(Y) = Y$), the maximal non-degenerate decomposition $\mathbf{D}(Y, R)$ is unique (see Theorem 1.3.11 of Laslier [1997]). Moreover, the *quotient ranking* $R/\mathbf{D}(Y, R)$ induces a pairwise-ranking on the components of $\mathbf{D}(Y, R)$. Formally, the global structure of a state is determined by the maximal decomposition of the Condorcet set.

[3] To get a better intuition for this definition, note that the Condorcet set is a component of R. In particular, $\mathbf{C}(R, Y)$ is the *smallest component* of R such that $Y \backslash \mathbf{C}(R, Y)$ is also a component of R where cRy for some $c \in \mathbf{C}(R, Y)$ and $y \in Y \backslash \mathbf{C}(R, Y)$.

Definition 3 (Global Structure) *For a pairwise-ranking R on $Y \subseteq X$, the* ***global structure*** $\langle \mathbf{G}(Y), R_G \rangle$ *is a pair consisting of the maximal decomposition* $\mathbf{G}(Y) = \mathbf{D}(\mathbf{C}(Y), R)$ *of R on the Condorcet set and the quotient ranking* $R_G = R/\mathbf{D}(\mathbf{C}(Y), R)$. *Moreover, any component* $g \in \mathbf{G}(Y)$ *defines a* ***locale****.*

States R and R' are *globally distinct* if $\langle \mathbf{G}(X), R_G \rangle \neq \langle \mathbf{G}'(X), R'_G \rangle$ and *globally similar* if $\langle \mathbf{G}(X), R_G \rangle = \langle \mathbf{G}'(X), R'_G \rangle$. In other words, two rankings are similar if they have the same Condorcet set $\mathbf{C}$ and the global structure of the rankings on $\mathbf{C}$ is similar. Conversely, two rankings are distinct when their Condorcet sets differ *or* the global structure of the rankings on $\mathbf{C}$ is distinct.

The condition for implementation on pairs of states can be stated in terms of the global structure. In particular, x and x' are said to be *pairwise implementable* on $Y \subseteq X$ (in states R and R') if there exists a voting agenda T on Y such that $c_T(R) = x$ and $c_T(R') = x'$. Given the main result, the following proposition fully characterizes implementation by agenda voting:

Proposition 1 (Pairwise Condition) ***(I)*** *For globally distinct states R and R', the outcomes x and x' are pairwise implementable iff $x \in \mathbf{C}(X, R)$ and $x' \in \mathbf{C}(X, R')$.* ***(II)*** *For globally similar states R and R', the outcomes x and x' are pairwise implementable iff they are in the same locale $g \in \mathbf{G}(X)$ and are pairwise implementable for some subset $g^* \subseteq g$.*

It is worth clarifying that the main result does not depend on the fact that voting is by majority. More generally, the pairwise-ranking on any profile $P \in \mathcal{P}$ may be derived from any *strong proper simple (SPS) game* (A, W). For a simple game (A, W), the set $W \subseteq 2^a$ defines a *monotonic* collection of winning coalitions such that $w \in W$ and $w \subseteq w'$ imply $w' \in W$. The simple game (A, W) is said to be strong and proper if a coalition w wins whenever its complement $A \backslash w$ loses (so that $w \in W$ iff $A \backslash w \notin W$).

Formally, any SPS game induces a pairwise ranking P_W such that $x P_W y$ iff $\{a \in A : x \succ_a y\} \in W$. When $\mathcal{P}_W \subseteq \mathcal{R}$ is the collection of pairwise rankings induced by the SPS game (A, W), then $f : \mathcal{P}_W \to X$ defines a (partial) Condorcet social choice function. Generalizing the notion of implementation defined above, a social choice function $F : \mathcal{L}^a \to X$ is said to be *implementable* if there exists an SPS game (A, W) and a voting agenda T such that $c_T(P_W) = F(P)$ for any profile $P \in \mathcal{P}$. Thus, F can be implemented by the SPS game (A, W) if and only if f_W is implementable and $F(P) = f_W(P_W)$ for any profile $P \in \mathcal{P}$. As such, the following generalization of the main result is immediate:

Theorem 1 *An SCF F is implementable if there exists an SPS game (A, W) and a (partial) Condorcet social choice function f s.t. (i) f is implementable for every pair of states and, (ii) $F(P) = f(P_W)$ for any $P \in \mathcal{P}$.*

One benefit of using an SPS game different from majority voting is that distinct outcomes may be implemented on profiles whose majority relations coincide. However, it should be kept in mind that departures from majority voting come at the cost of anonymity.

3 Discussion

In earlier work, Srivastava and Trick [1996] showed a necessary and sufficient condition for pairwise implementation on *adjacent states* R and R' that differ only on the pairwise-ranking of two outcomes y and y'. To state their condition, let $B_{R,R'}$ define the smallest component of R such that $\{y, y'\} \subseteq B_{R,R'}$. Srivastava and Trick established that *distinct* outcomes are pairwise implementable on R and R' iff they are in the Condorcet sets of X *and* $B_{R,R'}$ for each state.

Proposition 2 (Adjacent Pairwise Implementation) *x and x' are pairwise implementable on adjacent states R and R' iff: (i) $x \in \mathbf{C}(R, X)$ and $x' \in \mathbf{C}(R', X)$; and, (ii) $x = x'$ or, $x \in \mathbf{C}(R, B_{R,R'})$ and $x' \in \mathbf{C}(R', B_{R,R'})$.*

When two states are adjacent, the *pairwise condition* in Proposition 1 reduces to that given in Proposition 2. While the pairwise condition is more complex than the condition given in Proposition 2, it is somewhat easier to interpret.

For globally distinct states, pairwise implementation is virtually unrestricted. It is sufficient that the outcomes are drawn from the Condorcet sets.[4] Since agenda voting always draws from the Condorcet set, this requirement is more generally necessary for implementation (see e.g. Moulin [1986], Lemma 9).

For globally similar states, there are stronger restrictions on what can be implemented. As is more generally necessary for implementation in this environment, the outcomes must be drawn from the same locale g (see e.g. Moulin [1986], Lemma 10). Within any given locale however, the restrictions are relatively weak. Any pair of outcomes that can be implemented on a subset g^* of g can also be implemented on X.

The proof of the main result follows directly from Proposition 1. The result is obtained by algebraic methods and does not rely on the explicit construction of a mechanism (i.e. an agenda). Since the approach is somewhat unconventional, it will be helpful to provide a brief overview.

To get the basic intuition, first consider any collection $\mathcal{R}^d$ of globally distinct states. Now, fix a pair of states $R_j, R_k \in \mathcal{R}^d$ and a pair of outcomes $x_j \in \mathbf{C}(X, R_j)$ and $x_k \in \mathbf{C}(X, R_k)$. From Proposition 1(I), there exists an agenda $T(j,k)$ such that $c_{T(j,k)}(R_j) = x_j$ and $c_{T(j,k)}(R_k) = x_k$. Let $\mathcal{T}^d$ define any collection of agendas $T(j,k)$ ranging over all pairs of outcomes $x_j \in \mathbf{C}(X, R_j)$, $x_k \in \mathbf{C}(X, R_k)$ and all pairs of states $R_j, R_k \in \mathcal{R}^d$. Let $\mathcal{C}(\mathcal{T}^d)$ define the collection of choice functions c_T that correspond to some $T \in \mathcal{T}^d$. Any agendas T_1 and T_2 in $\mathcal{T}^d$ may be joined at the root to obtain a new agenda $T_1 + T_2$ and a new agenda choice function $c_{T_1+T_2}$ such that

$$c_{T_1+T_2}(R) = \max_R \{c_{T_1}(R), c_{T_2}(R)\}$$

Applying a theorem in universal algebra due to Maroti [2002], it can be shown that the closure of $\mathcal{C}(\mathcal{T}^d)$ under agenda concatenation coincides with the collection of agenda choice functions that select from the Condorcet set in every state $R \in \mathcal{R}^d$. Formally:

Proposition 3 (Globally Distinct States) *For any collection of globally distinct states $\mathcal{R}^d$, the (partial) Condorcet social choice function $f^d : \mathcal{R}^d \to X$ is implementable iff $f^d(R) \in \mathbf{C}(X, R)$ for all $R \in \mathcal{R}^d$.*

In other words, Proposition 3 shows that the condition in Proposition 1(I) is necessary and sufficient for globally distinct states. The next result shows the necessity and sufficiency of the condition in Proposition 1(II) for globally similar states.

Proposition 4 (Globally Similar States) *For any complete collection of globally similar states $\mathcal{R}^s$, the (partial) Condorcet social choice function $f^s : \mathcal{R}^s \to X$ is implementable iff it is pairwise implementable for every pair of states $R, R' \in \mathcal{R}^s$.*

Like Proposition 3, the proof of this result leverages the algebraic structure of agendas. To get the basic intuition, consider any collection $\mathcal{R}^s$ of globally similar states with Condorcet set $\mathbf{C}$. Given an outcome $x \in \mathbf{C}$, it is not difficult to construct an agenda $T(x)$ that implements x

[4] Srivastava and Trick [1996] show the sufficiency of this condition when the Condorcet sets of R and R' are distinct (see Theorem 2 of their paper). It is easy to see that this is a corollary of Proposition 2.

for every $R \in \mathcal{R}^s$. To see this, suppose that $\mathbf{G}(X) = \{g_i\}_{i=1}^k$ is the maximal decomposition $\mathbf{C}$ so that $g_i R_G g_{i+1}$ for $i < k$ and $g_k R_G g_1$. To implement $x \in g_1$, construct an elimination agenda $T(x)$ using the list

$$L = (x, g_2, ..., g_k, g_1 \backslash \{x\}, X \backslash \mathbf{C})$$

At every node, append an agenda that contains every outcome in g_i (resp. $g_1 \backslash \{x\}$ and $X \backslash \mathbf{C}$). The fact that $c_{T(x)}(R) = x$ for every state $R \in \mathcal{R}^s$ is a simple extension of a result due to Miller [1977]. Define $\mathcal{T}^s = \{T(x) : x \in \mathbf{C}\}$ so that $\mathcal{C}(\mathcal{T}^s)$ describes a collection of agenda choice functions c_T that pick the same outcome for every $R \in \mathcal{R}^s$. Clearly, the collection $\mathcal{T}^s$ satisfies the *pairwise condition.* Proposition 4 then follows by establishing that the closure of $\mathcal{C}(\mathcal{T}^s)$ under agenda concatenation generates *all* of the social choice functions that satisfy the *pairwise condition.* Given Propositions 3 and 4, the main result then follows by induction on the number of globally distinct sub-collections in $\mathcal{R}$. The details are presented in Section 4.

4 Proofs

4.1 Proof of Proposition 1

Proposition 1(I) follows from Theorem 1 of Srivastava and Trick [1996]. The following definition is required to state this result: a subset $PS \subseteq X$ is *prime* if there is no non-trivial partition $\mathbf{PS} = \{PS_i\}_{i=1}^k$ of PS such that: (i) $\mathbf{PS}$ is a decomposition of R and R' on PS; and, (ii) the quotient relations induced by $\mathbf{PS}$ agree so that $R/\mathbf{PS} = R'/\mathbf{PS}$.

Theorem 1 of Srivastava and Trick *The outcomes x and x' are pairwise implementable on states R and R' for some subset of X iff there exists a prime set PS such that $\{x, x'\} \subseteq PS \subseteq X$, $x \in \mathbf{C}(PS, R)$, and $x' \in \mathbf{C}(PS, R')$.*

Proposition 1(I) is a consequence of the following lemma.

Lemma 1 *If $\langle \mathbf{G}(X), R_G \rangle \neq \langle \mathbf{G}'(X), R'_G \rangle$, $\mathbf{C}(X, R) \cup \mathbf{C}(X, R')$ is a prime set.*

Proof. Omitted due to lack of space. ■

Proof of Proposition 1(I). ($\Leftarrow$) By Lemma 1, $\mathbf{C} \cup \mathbf{C}'$ is a prime set. By Theorem 1 of Srivastava and Trick, any $x \in \mathbf{C} = \mathbf{C}(\mathbf{C} \cup \mathbf{C}', R)$ and $x' \in \mathbf{C}' = \mathbf{C}(\mathbf{C} \cup \mathbf{C}', R')$ are pairwise implementable for $\mathbf{C} \cup \mathbf{C}' \subseteq X$. To complete the proof, fix a pair $x \in \mathbf{C}$ and $x' \in \mathbf{C}'$ and an agenda T that implements (x, x') for $\mathbf{C} \cup \mathbf{C}'$. Next, construct an agenda whose left branch at the root corresponds with T and whose right branch is any agenda on $X \backslash (\mathbf{C} \cup \mathbf{C}')$. (When $X \backslash (\mathbf{C} \cup \mathbf{C}') = \emptyset$, the right branch can be omitted.) By construction, the desired outcome emerges from the left branch in each state and defeats whatever emerges from the right. ($\Rightarrow$) If x and x' are pairwise implementable, $x \in \mathbf{C}$ and $x' \in \mathbf{C}'$ (by Lemma 9 of Moulin [1986]). ■

Proposition 1(II) is a consequence of the following lemma:

Lemma 2 *Given a collection of globally similar states $\mathcal{R}^s$, the (partial) Condorcet social choice function $f^s : \mathcal{R}^s \to X$ is implementable iff f^s is implementable for some $g^* \subseteq g \in \mathbf{G}(X)$.*

Proof. For parsimony, let $\mathbf{C} = \mathbf{C}(X, R)$ and $\mathbf{G}(X) = \{g_i\}_{i=1}^k$. First, fix an element $x \in g_i$ and suppose that $g_i R_G g_{i+1}$ for $i < k$ and $g_k R_G g_1$ (otherwise, the components can be relabeled). Construct an elimination agenda $T(x)$ using $L = (x, ..., g_k, g_1 \backslash g\{x\}, X \backslash \mathbf{C})$. As in the proof of

Proposition 1(I), the bottom branch may be omitted when $X\backslash\mathbf{C}=\emptyset$. To the branch labelled x, append the item x. To the branches labelled by g_i (respectively $g_1\backslash\{x\}$ or $X\backslash\mathbf{C}$), append an agenda T_i containing every outcome in g_i (resp. $g_1\backslash\{x\}$ or $X\backslash\mathbf{C}$). By construction, $T(x)$ implements x on $\mathcal{R}^s$ (see e.g. Lemma 8.3.3 of Laslier [1997]). Moreover, it can be associated with the trivial agenda $t(x)=x$ that implements x on $\{x\}\subseteq g_1$.

Let $\mathcal{T}_1=\{T(x) : x \in \mathbf{C}\}$ define the collection of agendas $T(x)$ on $\mathbf{C}$. Similarly, let $\mathcal{T}_1(g^*)=\{t(x) : x \in g^*\}$ define the collection of agendas $t(x)$ on $g^* \subseteq g \in \mathbf{G}(X)$. By construction, any agenda-implementable f^s on X can be obtained by concatenating agendas in $\mathcal{T}_1$. Since $f^s(R) \in \mathbf{C}$ for all $R \in \mathcal{R}^s$ (by Lemma 9 of Moulin [1986]), one can ignore agendas $T(x)$ where $x \notin \mathbf{C}$. Likewise, any agenda-implementable f^s on $g^* \subseteq g$ can be obtained by concatenating agendas in $\mathcal{T}_1(g^*)=\{x : x \in g^*\}$.

Define $\mathcal{T}_n=\{T_{n-1}+T_k : T_{n-1}\in\mathcal{T}_{n-1}$ and $T_k \in \mathcal{T}_k$ for $k<n\}$ and let $\mathcal{C}_n=\{f^s{:}f^s=c(T_n)$ for some $T_n \in \mathcal{T}_n\}$ (where $c(T_n)$ is the Condorcet social choice function implemented by T_n). Likewise, let $\mathcal{T}_n(g^*)=\{t_{n-1}(g^*)+t_k(g^*) : t_{n-1}(g^*)\in\mathcal{T}_{n-1}(g^*)$ and $t_k(g^*)\in\mathcal{T}_k(g^*)$ for $k<n\}$ and let $\mathcal{C}_n(g^*)=\{f^s{:}f^s=c(t_n)$ for $t_n\in\mathcal{T}_n(g^*)\}$.

Using strong induction, I establish: $f^s=c(T_n)\in\mathcal{C}_n$ iff $f^s=c(t_n(g^*))\in\mathcal{C}_n(g^*)$ for some $g^*\subseteq g\in\mathbf{G}(X)$. The claim is trivial for the base case $n=1$. So, suppose it holds for $n\le N$.

($\Rightarrow$) Now, consider any $f^s=c(T_N+T_k)\in\mathcal{C}_{N+1}(\mathcal{T})$. By the induction step, $c(T_N)=c(t_N(g_1^*))$ for some $t_N(g_1^*)$ on $g_1^*\subseteq g_1$ and $c(T_k)=c(t_k(g_2^*))$ for some $t_k(g_2^*)$ on $g_2^*\subseteq g_2$. There are two cases: (i) $g_1\neq g_2$; and, (ii) $g_1=g_2$. (i) Suppose, without loss of generality, that $g_1(R_G)g_2$. Then:

$$f^s=c(T_N+T_k)=c(T_N)+c(T_k)=c(T_N)=c(t_N(g_1^*))$$

where $t_N(g_1^*)$ implements f^s on $g_1^*\subseteq g_1\in\mathbf{G}(X)$ (by the induction step). (ii) In this case:

$$f^s=c(T_N+T_k)=c(T_N)+c(T_k)=c(t_N(g_1^*))+c(t_k(g_2^*))=c(t_N(g_1^*)+t_k(g_2^*))$$

where $t_N(g_1^*)+t_k(g_2^*)$ implements f^s on $g_1^*\cup g_2^*\subseteq g_1\in\mathbf{G}(X)$.

($\Leftarrow$) Suppose $f^s=c(t_N(g_1^*)+t_k(g_2^*))\in\mathcal{C}_{N+1}(g^*)$ for $t_N(g_1^*)$ on $g_1^*\subseteq g^*\subseteq g$ and $t_k(g_2^*)$ on $g_2^*\subseteq g^*\subseteq g$. By the induction step, $c(t_N(g_1^*))=c(T_N)$ and $c(t_k(g_2^*))=c(T_k)$ for $c(T_N)\in\mathcal{C}_N$ and $c(T_k)\in\mathcal{C}_k$. Following the same reasoning as case (ii) above, $f^s=...=c(T_N+T_k)$ where T_N+T_k implements f^s. ■

Proof of Proposition 1(II). Given Lemma 2, let $\mathcal{R}^s=\{R,R'\}$. ■

4.2 Proofs of Proposition 3, Proposition 4, and the Main Result

The proofs of these results rely on algebraic methods. Some preliminary definitions are required.

4.2.1 Preliminaries

Given a pairwise-ranking R on X, let the *tournament algebra* $\mathbf{X}$ be defined by a pair $(X,+)$ consisting of X and a binary operation $+$ such that $x+y=x$ iff xRy or $x=y$.[5] In turn, tournament algebras can be extended to products. Given a collection $\{\mathbf{X}_i\}_{i=1}^m$ of tournament algebras, the *product algebra* $\Pi_{i=1}^m\mathbf{X}_i$ is defined by $(\Pi_{i=1}^m X_i,+)$ where $+$ applies the operations $+_i$ component-wise so that $x+y\equiv(x_i+_i y_i)_{i=1}^m$. The projection of $x\equiv(x_i)_{i=1}^m\in\Pi_{i=1}^m X_i$ onto any collection $J\subseteq\{1,...,m\}$ of components is $\pi_J(x)=\Pi_{i\in J}x_i$. A *subdirect product* of $(\Pi_{i=1}^m X_i,+)$ is a sub-algebra $\mathbf{Y}\equiv(Y,+)$ of $\Pi_{i=1}^m\mathbf{X}_i$ (i.e. $Y\subseteq\Pi_{i=1}^m X_i$ and Y is closed under the

[5] More generally, an algebra $\mathbf{X}$ is a set X that is algebraically closed under a collection of n-ary operations.

binary operation $+$) such that $Y_i \equiv \{\pi_{\{i\}}(y) : y \in Y\} = X_i$ for any component Y_i. The subdirect product $\mathbf{Y}$ is *weakly indecomposable* if there exists no bi-partition (J, K) of the m components such that $Y = \pi_J(Y) \times \pi_K(Y)$ (up to re-ordering of the components).

A tournament algebra $(X, +)$ is *cyclic* if $\mathbf{C}(X, R) = X$ where R is the relation induced by the binary operation $+$ (so that xRy iff $x + y = x$ and $x \neq y$). A *congruence* β on $\mathbf{Y} \equiv (Y, +)$ is an equivalence relation on Y such that $(x + y)\beta(x' + y')$ iff $x\beta x'$ and $y\beta y'$. The largest congruence on $\mathbf{Y}$ is the *complete relation* $\mathbf{1}_Y = Y \times Y$ while the smallest is the *trivial relation* $\mathbf{Id}_Y = \{(y, y) : y \in Y\}$. Given a congruence β on $\mathbf{Y}$, the *quotient algebra* $\mathbf{Y}/\beta$ is $(Y/\beta, +_\beta)$ where Y/β is the partition of Y induced by β and $+_\beta$ is the binary operation $y/\beta +_\beta y'/\beta \equiv \{Z \in Y/\beta : y + y' \in Z\}$. Finally, $\mathbf{Y}$ is *irreducible* when its only congruences are $\mathbf{1}_Y$ and $\mathbf{Id}_Y$.

4.2.2 Proofs

The proofs of these results rely on a theorem in universal algebra established by Maroti [2002] (combining Lemmas 5.10 and 5.14 of his Ph.D. dissertation). To state Maroti's theorem:

Theorem (Maroti) *Let Y be a weakly indecomposable subdirect product of m cyclic tournament algebras. Then, Y has a unique largest congruence $\beta \neq Y \times Y$ and Y/β is an irreducible tournament algebra.*

They also rely on the following:

Claim 1 ***(I)*** *Natural numbers h and $h + 1$ are co-prime.* ***(II)*** *If a and b are co-prime, then every pair of congruence relations of the form $x = k(\bmod\, a)$ and $x = l(\bmod\, b)$ has a solution.*

Proof. Omitted due to lack of space. ■

To simplify the presentation below, consider the following definitions. Let $\mathcal{R}(X) = \{R_i\}_{i \in I}$ denote the collection of states on X. For parsimony, I abbreviate $\mathbf{C}(X, R_i)$ to $\mathbf{C}_i$. If there are n outcomes, denote the domain by X_n so that $\mathcal{R}(n)$ defines the collection of states on X_n. Let $\mathcal{R}^d_J(X) = \{R_j\}_{j \in J}$ denote a collection of $J \subseteq I$ globally distinct states in $\mathcal{R}(X)$ so that $\mathcal{R}^d(n)$ denotes any maximal collection of globally distinct states in $\mathcal{R}(n)$. Let $\mathcal{R}^s_j(n)$ denote the maximal collection (or *class*) of states that are globally similar to $R_j \in \mathcal{R}^d(n)$ and let $K(j) \subseteq I$ denote the set of indices associated with $\mathcal{R}^s_j(n)$. Finally, let $\mathbf{R}(n) = \{\mathcal{R}^s_j(n)\}_{j \in J}$ denote the partition dividing $\mathcal{R}(n)$ into classes of globally similar states.

It is possible to identify any Condorcet social choice function $c : \mathcal{R}(X) \to X$ with a vector $\vec{x} \equiv (x_i)_{i \in I} \in \Pi_{i \in I} X$. Using this approach, let $\mathcal{C}(n) = \{\vec{x} \in \Pi_{i \in I}\mathbf{C}_i : \vec{x} \text{ is implementable}\}$ denote the collection of agenda-implementable Condorcet social choice functions on X_n. Let $\mathcal{C}^d_J(X) = \{\pi_J(\vec{x}) \in \Pi_{j \in J}\mathbf{C}_j : \vec{x} \in \mathcal{C}(X)\}$ denote the collection of agenda-implementable Condorcet social choice functions on $\mathcal{R}^d_J(X)$. And, let $\mathcal{C}^s_j(n) = \{\pi_{K(j)}(\vec{x}) \in \Pi_{k \in K(j)}\mathbf{C}_k : \vec{x} \in \mathcal{C}(n)\}$ denote the collection of agenda-implementable Condorcet social choice functions on $\mathcal{R}^s_j(n) = \{R_k\}_{k \in K(j)}$.

Proof of Proposition 3. ($\Rightarrow$) If $f^d : \mathcal{R}^d \to X$ is implementable, it is also pairwise implementable for every pair of states in $\mathcal{R}^d$. From Proposition 1(I), $f^d(R) \in \mathbf{C}$ for all $R \in \mathcal{R}^d$.

($\Leftarrow$) Let $\mathcal{R}^d_I = \{R_i\}_{i \in I}$ and suppose that $|\mathbf{C}_i| > 1$. To establish the result, I show $\mathcal{C}^d_I(X) = \Pi^I_{i=1}\mathbf{C}_i$. The proof is by induction on the number of globally distinct states I. Proposition 1(I) proves the base case $I = 2$. Assume that the result holds for $|I| = n$. To complete the induction, I show the result $|I| = n + 1$. To simplify the notation, let $\bar{X} \equiv \Pi^{n+1}_{i=1}\mathbf{C}_i$ and $Y \equiv \mathcal{C}_{n+1}(\mathcal{T}^d)$ so that $\bar{X}_J = \pi_J(\bar{X})$ and $Y_J = \pi_J(Y)$ define the projections onto the sub-collection of states in J. To establish $Y = \bar{X}$, suppose otherwise.

First, note that Y is a subdirect product of $\bar{X}$. By the induction hypothesis, $\mathcal{C}^d_{J(n)}(X) = \Pi_{i\in J(n)}\mathbf{C}_i$ for any collection $J(n)$ of n states. Accordingly, $\pi_i(\mathcal{C}^d_{J(n)}(X)) = \mathbf{C}_i$. Second, each component of Y is cyclic because $Y_i = \mathbf{C}_i$. Finally, Y is weakly indecomposable. To see this, suppose $Y = \pi_J(Y) \times \pi_K(Y)$. By the induction step, $\pi_J(Y) = \Pi_{j\in J}\mathbf{C}_j$ and $\pi_K(Y) = \Pi_{k\in K}\mathbf{C}_k$ so that $Y = \Pi_{j\in J}\mathbf{C}_j \times \Pi_{k\in K}\mathbf{C}_k = \bar{X}$. But this contradicts the assumption that $Y \neq \bar{X}$ and establishes Y is weakly indecomposable. As such, Maroti's theorem applies. Let β define the largest congruence of Y such that $\beta \neq Y \times Y$. There are two cases to consider: (i) $|\bar{X}_j| = |\bar{X}_k| = h+1 > 1$ for all $j, k \leq n+1$; and (ii) there are distinct states j and k such that $|\bar{X}_j| \neq |\bar{X}_k|$.

(i) Pick any two states j and k and consider any distinct a, $b \in Y$. Label the elements of $\bar{X}_j$ so that the sequence $\{x^l_j\}^{h+1}_{l=0}$ defines a complete cycle $x^0_j R_j ... R_j x^l_j R_j ... R_j x^{h+1}_j = x^0_j$ in $\bar{X}_j$. And, label the elements of $\bar{X}_k$ so that $\{x^m_k\}^{h+1}_{m=0}$ defines a complete *reverse cycle* $x^0_k = x^{h+1}_k R_k ... R_k x^m_k R_k ... R_k x^0_k$ in $\bar{X}_k$. By the base case, there is a $x^{(l,m)}_{-jk} \in \Pi_{i\in I\setminus\{j,k\}}X_i$ s.t. $x^{(l,m)} \equiv (x^l_j \times x^m_k \times x^{(l,m)}_{-jk}) \in Y$. Without loss of generality, let $a \equiv x^{(0,0)}$. By construction, $x^{(l,m)}$ and $x^{(l+1,m+1)}$ are unranked by $\Pi^{N+1}_{i=1}R_i$. Since Y/β is a tournament, $(x^{(l,m)}, x^{(l+1,m+1)}) \in \beta$ for $l \leq h$ and $m \leq h$ so that $(a, x^{(l+1,m+1)}) \in \beta$.

By Theorem 7 of Harary and Moser [1966], there exists an h-length cycle $C_j \subseteq X_j$ containing b_j. Let l^* be the lowest index l such that $x^l_j \in C_j$ and let $x^* = x^{(l^*,l^*)}$. So, it is possible to label the elements of C_j so that the sequence $\{x^l_j\}^h_{l=0}$ defines a complete cycle $x^{l^*}_j = x^0_j R_j ... R_j x^l_j R_j ... R_j x^h_j = x^0_j$ in C_j. Because h and $h+1$ are co-prime, $(x^{(l,m)}, x^{(l',m')}) \in \beta$ for any l, $l' \leq h$ and m, $m' \leq h+1$ (by Claim 1). In particular, $(x^*, b) \in \beta$. Since $(a, x^*) \in \beta$ (by the first argument), it then follows that $(a, b) \in \beta$ so that $\beta = Y \times Y$.

(ii) Fix components j and k such that $|\bar{X}_j| = h' > h = |\bar{X}_k|$ and consider any distinct a, $b \in Y$. By the same approach as in the previous case, define a complete cycle on $\bar{X}_j$ and a complete reverse cycle on $\bar{X}_k$ such that a corresponds to the first element in each sequence. By Theorem 7 of Harary and Moser [1966], there exists an $(h+1)$-length cycle $C_j \subseteq X_j$ that contains b_j. Let l^* be the lowest index l such that $x^l_j \in C_j$ and let $x^* = x^{(l^*,l^*)}$. By the same argument given in the previous case, $(a, x^*) \in \beta$ and $(x^*, b) \in \beta$ so that $(a, b) \in \beta$ so that $\beta = Y \times Y$.

In both cases, $\beta = Y \times Y$ follows from $Y \neq \bar{X}$. But this contradicts $\beta \neq Y \times Y$. Thus, $Y = \bar{X}$. Given any collection of distinct states $\mathcal{R}^d$, it then follows that f^d is implementable if $f^d(R_j) \in \mathbf{C}_j$ for all $R_j \in \mathcal{R}^d$. The proof covers $\mathcal{R}^d$ consisting of *non-trivial* states such that $|\mathbf{C}_j| > 1$. This is sufficient to establish the result for any collection of distinct states $\mathcal{R}^d$. ■

The following lemma is needed in the proof of Proposition 4:

Lemma 3 *Given a complete collection of globally similar states $\mathcal{R}^s$, the (partial) Condorcet social choice function $f^s : \mathcal{R}^s \to X$ is implementable for every pair of states in $\mathcal{R}^s$ iff f^s is implementable for every pair of states on a subset g^* of some $g \in \mathbf{G}(X)$.*

Proof. Let $\mathcal{PW}(n) = \{\vec{x} \in \Pi_{i\in I}\mathbf{C}_i : \vec{x}$ satisfies the *pairwise condition* on $\mathcal{R}(n)\}$ represent the collection of Condorcet social choice functions that are pairwise implementable on X_n. Now, consider the similarity class $\mathcal{R}^s(n) = \{R_k\}_{k\in K}$ with global structure $\mathbf{G}(X_n) = \{g_l\}_{l\in L}$. Let $\mathcal{PW}^s(n) = \{\pi_K(\vec{x}) \in \Pi_{k\in K}\mathbf{C}_k : \vec{x} \in \mathcal{PW}(n)\}$ represent the choice functions that are pairwise implementable on $\mathcal{R}^s(n)$. First note that:

$$\mathcal{PW}^s(n) = \bigcup_{l\in L} \mathcal{PW}^s_l(n)$$

where $\mathcal{PW}^s_l(n) = \{\pi_K(\vec{x}) \in \Pi_{k\in K}\mathbf{C}_k : \vec{x} \in \mathcal{PW}^s(n) \cap \Pi_{k\in K}g_l\}$ is the sub-collection of $\mathcal{PW}^s(n)$ selecting from $g_l \in \mathbf{G}(X_n)$. To see this, fix adjacent states R and R' in $\mathcal{R}^s(n)$ such that

$f^s(R) = x \in g_l$ and $f^s(R') = x'$. By assumption, f^s is pairwise implementable for R and R'. From Proposition 1(II), $x \in g_l$ implies $x' \in g_l$. By the same argument, $f^s(R'') \in g_l$ for all $R'' \in \mathcal{R}^s(n)$.

Let $\mathcal{PW}_l^s(n)|_{g^*}$ define the sub-collection of $\mathcal{PW}_l^s(n)$ that is pairwise implementable on $g^* \subseteq g_l$. And, let $\mathcal{PW}_l^s(n)[g^*]$ define the sub-collection of $\mathcal{PW}_l^s(n)$ with range $g^* \subseteq g_l$ (so that $\cup_{k\in K}\{f^s(R_k)\} = g^*$ for any $f^s \in \mathcal{PW}_l^s(n)[g^*]$). By construction, $\mathcal{PW}_l^s(n) = \bigcup_{g^*\subseteq g_l} \mathcal{PW}_l^s(n)[g^*]$. To establish the desired result, it suffices to prove $\mathcal{PW}_l^s(n)|_{g^*} = \mathcal{PW}_l^s(n)[g^*]$ for any $g^* \subseteq g_l$. Using this identity, it follows that

$$\mathcal{PW}^s(n) = \bigcup_{l\in L} \bigcup_{g^*\subseteq g_l} \mathcal{PW}_l^s(n)|_{g^*}$$

as required. To show $\mathcal{PW}_l^s(n)|_{g^*} = \mathcal{PW}_l^s(n)[g^*]$ for any $g^* \subseteq g_l$, first consider the following:

Claim A *If $f^s \in \mathcal{PW}_l^s(n)$, $\{R, R'\} \subseteq \mathcal{R}^s(n)$, and $\mathbf{C}(g_l, R) = \{f^s(R')\}$, then $f^s(R) = f^s(R')$.*

Proof. Omitted due to lack of space. ■

The result follows by establishing that $\mathcal{PW}_l^s(n)[g^*] \subseteq \mathcal{PW}_l^s(n)|_{g^*}$. The inverse inclusion $\mathcal{PW}_l^s(n)|_{g^*} \subseteq \mathcal{PW}_l^s(n)[g^*]$ follows from the fact that $f^s(R) = x$ for any $R \in \mathcal{R}^s(n)$ such that xRx' for all $x' \in g^*\backslash\{x\}$ (by Lemma 9 of Moulin [1986]). To establish $\mathcal{PW}_l^s(n)[g^*] \subseteq \mathcal{PW}_l^s(n)|_{g^*}$, there are two cases to consider: (i) $g^* = g_l$; and, (ii) $g^* \subsetneq g_l$.

(i) For globally distinct states such that $\mathbf{G}(g_l, R) \neq \mathbf{G}(g_l, R')$, it is sufficient to show that $f^s(R) \in \mathbf{C}(g_l, R)$ for all $R \in \mathcal{R}^s(n)$. To see this, consider $f^s \in \mathcal{PW}_l^s(n)[g_l]$ and fix some state R such that $|\mathbf{C}(g_l, R)| > 1$ and any $x' \in \mathbf{C}(g_l, R)$. (The fact that $f^s(R) \in \mathbf{C}(g_l, R)$ for any R such that $|\mathbf{C}(g_l, R)| = 1$ follows from Claim A and the assumption that $f^s \in \mathcal{PW}_l^s(n)[g_l]$.) Consider the state R' such that $R'|_{X\backslash g_l} = R|_{X\backslash g_l}$, $R'|_{g_l\backslash\{x'\}} = R|_{g_l\backslash\{x'\}}$, and $x'R'x$ for any $x \in g_l\backslash\{x'\}$. Since $f^s \in \mathcal{PW}_l^s(n)[g_l]$, x' is chosen for some state $R'' \in \mathcal{R}^s(n)$. By Claim A, it follows that $f^s(R') = x'$. By construction, $\{x'\} \subseteq PS \subseteq \mathbf{C}(g_l, R)$ for any non-trivial prime set PS on R and R'. By Theorem 1 of Srivastava and Trick, it then follows that $f^s(R) \in \mathbf{C}(g_l, R)$. This establishes $f^s(R) \in \mathbf{C}(g_l, R)$ for all $R \in \mathcal{R}^s(n)$.

Next, consider globally similar states such that $\mathbf{G}(g_l, R) = \mathbf{G}(g_l, R') = \{g_l^i\}_{i\in I}$. Without loss of generality, suppose $f^s(R) \in g_l^i$. It is sufficient to show that $f^s(R)$ and $f^s(R')$ are pairwise implementable for some $g \subseteq g_l^i$. From Theorem 1 of Srivastava and Trick, $f^s(R)$ and $f^s(R')$ are pairwise implementable for some prime set PS such that $f^s(R) \in PS$. By definition, it must be that $PS \subseteq g_l^i$ for any prime set such that $f^s(R) \in PS$. This establishes the desired result.

(ii) Pick $f^s \in \mathcal{PW}_l^s(n)[g^*]$ for some $g^* \subsetneq g_l$. Fix a state R and consider the state $R^{\downarrow g^*}$ defined by $R^{\downarrow g^*}|_{X\backslash g^*} = R|_{X\backslash g^*}$, $R^{\downarrow g^*}|_{g^*} = R|_{g^*}$, and $x'R^{\downarrow g^*}x$ for any $x' \in X\backslash g^*$ and any $x \in g^*$. By construction, any non-trivial prime set PS on R and $R^{\downarrow g^*}$ must contain some $x' \in X\backslash g^*$. Since $f^s(R)$ and $f^s(R^{\downarrow g^*})$ are pairwise implementable, $f^s(R) = f^s(R^{\downarrow g^*})$. Otherwise, $\mathbf{C}(PS, R^{\downarrow g^*}) \subseteq X\backslash g^*$ so that $f^s(R^{\downarrow g^*}) \in X\backslash g^*$ which contradicts the assumption that $f^s \in \mathcal{PW}_l^s(n)[g^*]$. This establishes that $f^s(R) = f^s(R')$ for any states R and R' in $\mathcal{R}^s(n)$ such that $R|_{g^*} = R'|_{g^*}$.

To see that $f^s(R) \in \mathbf{C}(g^*, R|_{g^*})$ for any $R \in \mathcal{R}_j^s(n)$, fix a state $\bar{R}$ such that $x\bar{R}x'$ for any $x \in g^*$ and $x' \in g_l\backslash g^*$. By the same reasoning as in (i) above, $f^s(\bar{R}) \in \mathbf{C}(g^*, \bar{R})$. Since $f^s(R) = f^s(\bar{R})$ for any R and $\bar{R}$ in $\mathcal{R}^s(n)$ such that $R|_{g^*} = \bar{R}|_{g^*}$, then $f^s(R) \in \mathbf{C}(g^*, R)$ for all $R \in \mathcal{R}^s(n)$.

To complete the proof, fix any state R and consider the state $R^{\uparrow g^*}$ defined by $R^{\uparrow g^*}|_{X\backslash g^*} = R|_{X\backslash g^*}$, $R^{\uparrow g^*}|_{g^*} = R|_{g^*}$, and $xR^{\uparrow g^*}x'$ for any $x \in g^*$ and $x' \in X\backslash g^*$. Now consider any R' globally similar to R on g^*. By construction, $R^{\uparrow g^*}|_{g^*} = R|_{g^*}$ and $R'^{\uparrow g^*}|_{g^*} = R'|_{g^*}$ so that $f^s(R) = f^s(R^{\uparrow g^*})$ and $f^s(R') = f^s(R'^{\uparrow g^*})$. Moreover, $R^{\uparrow g^*}$ and $R'^{\uparrow g^*}$ are globally similar on g^*. Without loss of generality, suppose that $\mathbf{G}(g^*, R) = \mathbf{G}(g^*, R') = \{g_i^*\}_{i\in I}$ and $f^s(R) \in g_i^*$.

Following the same reasoning as in (i) above, $f^s(R^{\uparrow g^*})$ and $f^s(R'^{\uparrow g^*})$ are pairwise implementable for some prime set $PS \subseteq g_i^*$, which establishes the desired result. ■

Proof of Proposition 4 and Main Result. ($\Rightarrow$) If $f : \mathcal{R} \to X$ (respectively $f^s : \mathcal{R}^s \to X$) is implementable, it is implementable for every pair of states in $\mathcal{R}$ (respectively $\mathcal{R}^s$).

($\Leftarrow$) As in Lemma 3, let $\mathcal{PW}(n)$ represent the choice functions that satisfy the pairwise condition on $\mathcal{R}(n)$ and let $\mathcal{PW}_j^s(n)$ represent the choice functions that satisfy the pairwise condition on the similarity class $\mathcal{R}_j^s(n) = \{R_k\}_{k \in K(j)}$. Finally, let $J(n) = |\mathbf{R}(n)|$ represent the number of similarity classes in $\mathcal{R}(n)$. For Proposition 4, I show **(I)** $\mathcal{C}_j^s(n) = \mathcal{PW}_j^s(n)$ for any $j \in J(n)$. For the main result, I show **(II)** $\mathcal{C}(n) = \Pi_{j \in J}\mathcal{C}_j^s(n)$ for any n. Results **(I)** and **(II)** establish $\mathcal{C}(n) = \Pi_{j \in J}\mathcal{PW}_j^s(n)$. Since $\mathcal{PW}(n) = \Pi_{j \in J}\mathcal{PW}_j^s(n)$ by Proposition 1, it follows that $\mathcal{C}(n) = \mathcal{PW}(n)$. The proof is by strong induction on the size of the domain n and the number of similarity classes $J(n)$.

For $n \in \{1, 2, 3\}$, it is easy to see that **(I)** and **(II)** hold. (For $n = 2$, there are 2 globally distinct states each consisting of a linear order. For $n = 3$, there are 8 states and 5 similarity classes (3 classes that consist of two linear orders each and 2 classes consisting of one cycle).

For all $m < n$, assume $\mathcal{C}(m) = \Pi_{j \in J(m)}\mathcal{C}_j^s(m)$ and $\mathcal{C}_j^s(m) = \mathcal{PW}_j^s(m)$ for all $j \in J(m)$. In order to complete the induction, it is enough to show that **(I)** and **(II)** hold for n.

(I) Consider any non-trivial class similarity $\mathcal{R}_j^s(n) \in \mathbf{R}(n)$ (so that $|\mathcal{R}_j^s(n)| > 1$ or, equivalently, $|\mathbf{G}^j(X_n)| > 1$). Wlog, suppose $\mathbf{G}^j(X) = \{g_l^j\}_{l \in L(j)}$ so that $|g_l^j| < n$. By Lemma 2:

$$\mathcal{C}_j^s(n) = \bigcup_{l \in L(j)} \bigcup_{g^* \subseteq g_l^j} \mathcal{C}_{jl}^s(n)|_{g^*}$$

where $\mathcal{C}_{jl}^s(n)|_{g^*}$ is the collection of Condorcet social choice functions that are implementable on $g^* \subseteq g_l^j$. Lemma 3 above establishes that:

$$\mathcal{PW}_j^s(n) = \bigcup_{l \in L(j)} \bigcup_{g^* \subseteq g_l^j} \mathcal{PW}_{jl}^s(n)|_{g^*}$$

By induction assumptions **(I)** and **(II)**, $\mathcal{C}_{jl}^s(n)|_{g^*} = \mathcal{PW}_{jl}^s(n)|_{g^*}$ for any $g^* \subseteq g_l^j$. Consequently, $\mathcal{C}_j^s(n) = \mathcal{PW}_j^s(n)$ which establishes the desired result.

(II) First, let $J^*(n) = \{j \in J(n) : |\mathbf{C}_j| > 1\}$. Given $\mathcal{C}_j^s(n) = \mathcal{PW}_j^s(n)$ for every $j \in J^*(n)$, the result follows by induction on J. For ease of notation, let $\pi_J(\mathcal{C}(n)) = \pi_J$. To establish the base case $J = \{1, 2\}$, suppose $\pi_{\{1,2\}} \neq \pi_1 \times \pi_2$. Note that $\pi_{\{1,2\}}$ is a subdirect product of $\pi_1 \times \pi_2$. For any state $R_j \in \mathcal{R}_j^s(n)$, there exists an agenda $T(x)$ that implements every outcome in $x \in \mathbf{C}_j$. (The construction is similar to that given in Lemma 2.) This observation also establishes that the sub-algebra on each state is cyclic. Finally, the assumption that $\pi_{\{1,2\}} \neq \pi_1 \times \pi_2$ implies that $\pi_{\{1,2\}}$ is weakly indecomposable. To see this, suppose that there are two disjoint collections $\mathcal{R}_U = \{R_u : u \in U\}$ and $\mathcal{R}_V = \{R_v : v \in V\}$ such that $\mathcal{R}_U \cup \mathcal{R}_V = \mathcal{R}_1^s(n) \cup \mathcal{R}_2^s(n)$ and $\pi_{\{1,2\}} = \pi_U(\mathcal{C}(n)) \times \pi_V(\mathcal{C}(n))$. Now, consider any $R_1, R_1' \in \mathcal{R}_1^s(n)$ and suppose that $R_1 \in \mathcal{R}_U$ and $R_1' \in \mathcal{R}_V$. It follows that it is possible to pairwise implement $x \in g$ and $x' \in g'$ for $g \neq g'$. This contradicts Proposition 1 and establishes $\mathcal{R}_1^s(n) \subseteq \mathcal{R}_U$ or $\mathcal{R}_1^s(n) \subseteq \mathcal{R}_V$. A similar argument shows $\mathcal{R}_2^s(n) \subseteq \mathcal{R}_U$ or $\mathcal{R}_2^s(n) \subseteq \mathcal{R}_V$. Since the collections $\mathcal{R}_U$ and $\mathcal{R}_V$ are non-trivial, then $\mathcal{R}_1^s(n) = \mathcal{R}_U$ and $\mathcal{R}_1^s(n) = \mathcal{R}_V$ without loss of generality. But, this contradicts the assumption that $\pi_{\{1,2\}} \neq \pi_1 \times \pi_2$ and establishes that $\pi_{\{1,2\}}$ is weakly indecomposable.

Accordingly, the theorem of Maroti applies. Let β define the largest congruence of Y such that $\beta \neq \pi_{\{1,2\}} \times \pi_{\{1,2\}}$. By Proposition 1, it is possible to pairwise implement (x_1, x_2) and (x_1', x_2') on $R_1 \in \mathcal{R}_1^s(n)$ and $R_2 \in \mathcal{R}_2^s(n)$ so that $x_1 R_1 x_1'$ and $x_2' R_2 x_2$. Using the same approach as in Proposition 3, it follows that $\beta = \pi_{\{1,2\}} \times \pi_{\{1,2\}}$. But, this contradicts the assumption that $\beta \neq \pi_{\{1,2\}} \times \pi_{\{1,2\}}$ and establishes that $\pi_{\{1,2\}} = \pi_1 \times \pi_2$ in the base case $J = \{1, 2\}$.

Now, assume that the result holds for $|J| = j$. In order to complete the induction, it suffices to show that the result holds for $|J| = j + 1$. Following the same line of argument as in the base case (and Proposition 3), the result $\pi_J = \Pi_{j \in J} \pi_j$ can be established by contradiction. This proves $\pi_{J^*(n)}(\mathcal{C}(n)) = \Pi_{j \in J^*(n)} \mathcal{C}^s_j(n)$. It then follows that $\mathcal{C}(n) = \Pi_{j \in J(n)} \mathcal{C}^s_j(n)$. ■

References

[1] Abreu & Sen (1990), "Subgame Perfect Implementation", *J, Econ. Theory* 50 285-299.

[2] Black (1958), *The Theory of Committees and Elections*, New York: Cambridge University Press.

[3] Dutta & Sen (1993), "Implementing Generalized Condorcet Social Choice Functions via Backward Induction", *Soc. Choice Welfare* 10 149–160.

[4] Farquharson (1969), *Theory of Voting*, New Haven: Yale University Press

[5] Gol' berg & Gourvitch (1986), "Secret and Extensive Dominance Solvable Voting Schemes", Moscow *mimeo.*

[6] Harary & Moser (1966), "The Theory of Round Robin Tournaments", *Am. Math. Mon.* 73(3) 231–246.

[7] Herrero & Srivastava (1992), "Implementation via Backward Induction", *J. Econ. Theory* 56 70–88.

[8] Laslier (1997), *Tournament Solutions and Majority Voting*, Berlin/Heidelburg: Springer Verlag.

[9] Maroti (2002), "The Variety Generated by Tournaments", Vanderbilt University Ph.D. Dissertation.

[10] McKelvey & Niemi (1978), "A Multistage Game Representation of Sophisticated Voting for Binary Procedures", *J. Econ. Theory* 18 1-22.

[11] Moore (1992), "Implementation, Contracts, and Renegotiation in Environments with Complete Information", in *Advances in Economic Theory: Sixth World Congress, Vol. 1* (J-J Laffont ed.), New York: Cambridge University Press.

[12] Moore & Repullo (1988), "Subgame Perfect Implementation", *Econometrica* 56 1191-1220.

[13] Moulin (1979), "Dominance-Solvable Voting Schemes", *Econometrica* 47 1337-1351.

[14] Moulin (1986), "Choosing from a Tournament", *Soc. Choice Welfare* 3 271–291.

[15] Miller (1977), "Graph-Theoretical Approaches to the Theory of Voting", *Am. J. Polit. Sci.* 21 769–803.

[16] Palfrey & Srivastava (1989), "Nash Implementation Using Undominated Strategies", *Econometrica* 59 479–401.

[17] Srivastava & Trick (1996), "Sophisticated Voting Rules: The Case of Two Tournaments", *Soc. Choice Welfare* 13 275–289.

[18] Trick (2006), "Small Binary Voting Trees", Carnegie Mellon University *mimeo.*

[19] Trick (2009), "An Operations Research Look at Voting", Algorithmic Decision Theory Conference (Venice) *mimeo.*

[20] Vartiainen (2007a), "Subgame Perfect Implementation: A Full Characterization", *J. Econ. Theory* 133 111-126.

[21] Vartiainen (2007b), "Subgame Perfect Implementation of Voting Rules via Randomized Mechanisms", *Soc. Choice Welfare* 29 353-367.

Sean Horan
Département des sciences économiques
Université du Québec à Montréal (UQAM)
315 Ste-Catherine est, Montréal, QC, Canada
Email: `horan.sean@uqam.ca`