Noname manuscript No.
(will be inserted by the editor)

Implementability over non-cardinal preferences

Matthias Köppe · Maurice Queyranne · Christopher Thomas Ryan

the date of receipt and acceptance should be inserted later

Abstract We develop a novel concept of non-manipulability of social choice functions over non-cardinal preferences, which draws on strong connections with recent work in mechanism design. This concept, termed full implementability, weakens the strong notion of strategy-proofness and allows for expressive preferences while avoiding impossibility results which plague much of social choice theory. Strategy-proofness of a social choice function f may be characterized as the non-existence of negative *arcs* in graphs associated with f. In contrast, full implementability relaxes this requirement and only requires the non-existence of negative *cycles*. We show that the Borda scoring rule is fully implementable when there are three outcomes, and no scoring rule is fully implementable when there are four or more outcomes. In addition, we explore "local" conditions involving 2-cycles which guarantee full implementability. This connects with recent studies on monotonicity and implementability in mechanism design. In the course of our development we settle an open question in mechanism design over discrete domains posed by Mu'Alem and Schapira.

Keywords Strategy proofness · Implementability · Mechanism design · Discrete domains.

1 Introduction

The problem of designing an allocation mechanism that induces truthful revelation of preferences is well-studied in the social and computer sciences. There are essentially two main branches, which differ in terms of how agent preferences are expressed and whether utility transfers are permitted. In the first branch, commonly known as social choice, agents express strict ordinal preferences over alternatives and utility transfers are disallowed. In the second branch, commonly known as mechanism design, agent preferences are expressed in transferable cardinal utilities.

M. Köppe
University of California, Davis, Department of Mathematics
E-mail: mkoeppe@math.ucdavis.edu

M. Queyranne
University of British Columbia, Sauder School of Business
E-mail: maurice.queyranne@sauder.ubc.ca

C.T. Ryan
University of Chicago Booth School of Business
E-mail: chris.ryan@chicagobooth.edu

When faced with a discrete set of alternatives, there is debate over whether agents more naturally form ordinal preferences (rankings) or cardinal utilities over the set of alternatives. Arguments in defense of an ordinal approach date back to Arrow (1970) and earlier in Robbins (1938). The common defense is one of simplicity, it is thought to be relatively easy for agents to rank alternatives. Carroll (2011) provides further justification for ordinal preferences in a setting with random allocations. However, much of the defense for ordinal preferences comes in the form of attacks on cardinal preferences. Criticisms mainly focus on philosophical issues with interpersonal comparisons of utility and the transferability of utility between agents and other stores of value, such as money Robbins (1938). Despite this, some authors put forward a spirited and logical defense of cardinal preferences, the most respected due to Harsanyi (1955). Indeed, in the mechanism design literature cardinal utilities are largely a non-controversial primitive. Artificial intelligence researchers have furthermore made the case that computational agents are more natural at assigning cardinal values to alternatives than ranking of alternatives (Procaccia and Rosenschein (2006)).

In this paper, we take a hybrid view of preference formation and explore implications for truth-telling and implementability. The aim is to build a bridge between the theories of social choice and mechanism design in order to give refined notions of implementability in the social choice setting. The search for social choice functions with desirable properties is hampered by a myriad of impossibility results, including the landmark Gibbard-Sattherthwaite theorem. The implication is that most social choice functions used in practice suffer from fundamental flaws, the most common being lack of strategy-proofness. Great effort is put into designing markets which respect strategy-proofness and limit the introduction of other undesirable properties, an illustrative example being school choice (Pathak (2011)). In our approach we take a social choice function as given, and ask how the introduction of monetary transfers could ensure truth-telling in agents. However, unlike in mechanism design, where agents are assumed to be able to declare cardinal types, we assume agents' preferences are first formed by ranking alternatives, and then assigned cardinal values via these rankings. A social choice function is thought desirable if in a wide range of cardinal assignments to rankings by the agents, transfers can be found which induce truth telling with transfers.

This weakened truth-telling notion is an alternative to the common approach to circumvent impossibility by *restricting* the set of allowable preferences, e.g., to single-peaked preferences, see Black (1948). In our approach we do not restrict preferences and instead avoid impossibility by introducing new notions of truth-telling, which we term full and partial implementability. Mathematically, strategy-proofness corresponds to the non-existence to negative *arcs* in appropriately defined graphs called *type graphs*. Full and partial implementability are related to negative *cycles* in similarly defined graphs.

As the names suggest, full implementability relates to the implementability of allocation rules in the mechanism design context. The idea is as follows: every ordinal preference over a set of alternatives can be linked to a family of "consistent" cardinal utilities that respect the underlying ordering, and possibly other desirable properties. We call a member of this family an *anointment*[1] of the underlying ordinal preferences. Each agent has a family of anointments which represent the possible cardinal utility values she may have over set of alternatives. A social choice function is *fully implementable* if, no matter the choice of anointment within each agent's family of anointments, the resulting mechanism design setting has transfers which induce truth-telling. This, in turn, corresponds to the non-existence of negative cycles in a parameterized family of "anointed" type graphs.

[1] When a bishop is turned into a cardinal, the process is sometimes called an *anointment*. We are turning ordinal preference levels into cardinal utilities, hence the terminology.

Our notion of implementability is particularly desirable in situations where a designer is contemplating migrating from a market without money, to one where monetary transfers are permitted. Indeed, suppose a designer wants to implement the social choice function f over ordinal preference levels but Gibbard-Sattertherwaite implies there is no strategy-proof way to do this. If the designer has the option of utilizing money transfers the question now is whether f can be implemented in the sense of mechanism design, over a wide range of cardinal utility functions of the agents. This question is precisely the object of our investigation. Thus, as a property of a social choice function over ordinal preferences, our notion of implementability is a measure of its suitability or robustness to cardinal settings with utility transfers. From the perspective of the agents, our model captures the following behavioral consideration. If an agent knows the social choice function is fully implementability then, whatever her true utility function is, there exists a payment rule where she has no incentive to lie. This serves a potential deterrent to lying even in the absence of strategyproofness.

Moreover, we view our concept of implementability as providing an alternate notion of "degree of manipulability" different from those presented in Kelly (1993) and Lepelley et al (2008) which are based on counting instances of violations of strategy-proofness across type graphs. As we have discussed, strategy-proofness corresponds to negative arcs in type graphs. We contend that certain negative arcs are more damaging to truth-telling behavior than others: indeed a *negative cycle* in a type graph not only fails to guarantee truth-telling in a setting with no monetary transfers, but it also rules out the possibility of truth telling behavior *for any* system of monetary transfers which faithfully represent the underlying ordinal preferences. Partial implementability reveals a degree to which negative arcs can be allowed and yet still maintain truth-telling behavior in a family of cardinal extensions. In this sense, we do not simply count the number of violation of strategy-proofness, but provide theory behind which kinds of violations are more impactful than others.

Our results

Let $\mathbf{X}$ be the family of anointment profiles that the designer believes captures the possible cardinal preferences of the agents. A minimal requirement on $\mathbf{X}$ is that cardinal utility assignments be strictly increasing in the underlying ordinal preference levels. We denote this minimally structured set by $\mathbf{I}$. When a social choice function is implementable (in the sense of mechanism design) for every anointment in $\mathbf{I}$, that social choice function is *fully implementable.* However, it is often useful to restrict the family of anointments and so we introduce the notion of partial implementability: a social choice function is partially implementable if there exists a non-empty strict subset $\mathbf{X}$ of $\mathbf{I}$ for which the social choice function is implementable (again in the sense of mechanism design).

A key result (Theorem 2) is that relaxing strategy-proofness to partial implementability assuages the curse of impossibility theorems. Theorem 2 shows that for any number of alternatives (greater than three) and any number of agents (greater than two) there exist social choice functions which are non-dictatorial, Pareto optimal and partially implementable. The partial implementability is non-trivial, in the sense that there is a cone of anointments for which the social choice function is implementable.

Section 3 provides a comprehensive treatment of the implementability properties of scoring rules. Theorem 3 shows how every scoring rule is at least partially implementable. In particular, when the anointment for each player is set equal to the score vector, the resulting social choice function is implementable (in the sense of mechanism design). In this setting, this gives an interpretation of anointments as being monetary values equal to the scores, which are then transferred among the players via payments. Knowing every scoring rule is at least partially implementable, the question then becomes which scoring rules are fully implementable. This is fully answered in Theorems 4 and Theorem 5. Theorem 4 provides a characterization of implementability in the three outcome case. In particular, it shows that for three outcomes only the Borda scoring rule is fully implementable. Theorem 5 shows that no scoring rule is fully implementable with 4 or more outcomes.

From a computational perspective, a naive attempt to test implementability requires checking the nonnegativity of all cycles in a parametric family of type graphs. In Section 4 we explore certain "local conditions" for implementability, whereby the non-existence of 2-cycles is sufficient to imply the non-existence of negative cycles, and thus implementability over a restricted set of anointments. In the course of our development we settle an open question in mechanism design over discrete domains posed in Mu'alem and Schapira (2008).

Related work

The debate between ordinal and cardinal preferences in social choice has received renewed attention in the artificial intelligence literature, because, as discussed above, computational agents are thought to more readily possess cardinal preferences than ordinal preferences. This consideration has given rise to a series of papers in computational social choice which views social choice theory from a cardinal lens (see Boutilier et al (2012), Caragiannis and Procaccia (2011) and Procaccia and Rosenschein (2006)). These studies start with the contention that agents possess cardinal utilities over alternatives which are then "embedded" into ordinal rankings. These ordinal rankings, then, are aggregated as in classical social choice to determine an alternative. Attention has been given to how to optimally embed cardinal utilities into ordinal rankings for a given social choice function, and conversely how to design optimal social choice functions for a given embedding, the objective being to minimize the discrepancy between a socially optimal allocation of goods and the allocation chosen by the social choice function.

Our work differs from this stream of literature in two dimensions. First, our primary concern is non-manipulability in social choice, a topic not yet tackled in the work above. Second, in our work, ordinal preferences are taken as primitive with cardinal utilities formed through assigning anointments. Thus we follow the more standard assumption that agents more easily form ordinal preferences, which are primitive to our model. Cardinal utilities only enter when the design of transfers for ensuring truthfulness is considered, as is also standard in the literature.

In addition, our study of local conditions relates to recent interest in the mechanism design literature (see for instance Ashlagi et al (2010) and Carroll (2012)), which explores the question of how and when local incentives imply global incentives in allocation settings. Most results in that literature concern convex type sets, in our approach our cardinal type sets are non-convex and is most similar to Mu'alem and Schapira (2008). Viewed from the perspective of mechanism design, we show how local incentive conditions may imply global conditions across a family of parametrically related mechanism design settings.

2 Full and partial implementability

Suppose n agents face a set A of at least two alternatives. Each agent rates the alternatives according to a finite set $[L] = \{1, 2, \dots, L\}$ of *preference levels* ($L \geq 2$) where higher levels correspond to more preferred alternatives. Agent i states his preference as a vector $t_i = (t_i(a))_{a \in A} \in T_i \subseteq [L]^A$. Vector t_i is agent i's *type* and set T_i her *type space*. A *social choice function* f maps each *type profile* $\mathbf{t} = (t_1, \dots, t_n) \in \mathbf{T} := T_1 \times \cdots \times T_n$ of stated preference vectors to an alternative in A.

Our preference model subsumes strict total orders on A, a standard model in the social choice literature but allows for ties. Indeed, simply set $L = |A|$ and disallow ties in each T_i. Thus each t_i is a permutation of $[L] = \{1, \dots, |A|\}$ and each type set T_i is the set $[L]!$ of all permutations of $[L]$. For example, the strict total order $a > b > c$ of $A = \{a, b, c\}$ is represented as type $t = (t(a), t(b), t(c)) = (3, 2, 1)$.

Certain properties of social choice functions are considered desirable and have been extensively studied in the literature (see for instance Gaertner (2009)). We focus on the *strategy-proofness* property. Phrased in our setting, a social choice function f is *strategy-proof* if for every agent i, every type profile $\mathbf{t} = (t_i, \mathbf{t}_{-i}) \in \mathbf{T}$ and every alternative type $u_i \in T_i$ we have $t_i(f(t_i, \mathbf{t}_{-i})) \geq t_i(f(u_i, \mathbf{t}_{-i}))$. That is, when f is strategy proof, no agent has an advantage in misrepresenting her type, whatever the stated preferences of the other agents. As is standard, $\mathbf{t}_{-i}$ is the subprofile formed by the stated preference vectors of i's *rivals*, i.e., all agents other than i.

A well-known negative result in social choice theory due to Gibbard (1973) and Satterthwaite (1975) is that whenever (i) the social choice function f is strategy-proof and Pareto optimal, (ii) the space of type profiles T contains all strict total orders on A (i.e., $T_i = [L]!$ for all agents i), and (iii) there are at least three alternatives, then f must be *dictatorial* — the undesirable property that there exists a single agent i such that f allocates to every type profile an alternative most preferred by agent i. There are numerous alternate formulations of this result, some slightly weaker and some slightly stronger. The overall message is the same: if you ask for too much, you end up in dictatorship.

Our concept is motivated in part by the following interpretation of strategy-proofness. For each agent i and rival type profile $\mathbf{t}_{-i}$ define a complete digraph $G_{i,\mathbf{t}_{-i}}$, called a *type graph*, with node set T_i. For a given social choice function f, the *sign* of arc (u, t) is *positive* if $t(f(t, \mathbf{t}_{-i})) > t(f(u, \mathbf{t}_{-i}))$, *negative* if $t(f(t, \mathbf{t}_{-i})) < t(f(u, \mathbf{t}_{-i}))$; and *zero* if $t(f(t, \mathbf{t}_{-i})) = t(f(u, \mathbf{t}_{-i}))$. Using these notions we can restate the strategy-proofness of a given social choice function in very simple terms: f is strategy-proof if and only if no type graph $G_{i,\mathbf{t}_{-i}}$ contains a negative arc.

We weaken strategy-proofness by allowing negative arcs but disallowing certain types of "negative" cycles (as defined below). The motivation for this approach comes from the mechanism design literature, where notions of truth-telling relate to cycles (see, e.g. Vohra (2011)). To connect the ordinal setting and cardinal setting we define mappings of preference levels to cardinal utility values as follows. An *anointment* for agent i is a strictly increasing function x_i from $[L]$ into $\mathbb{R}$, where $x_i(\ell)$ is interpreted as the cardinal utility associated with preference level ℓ. We restrict to *strictly* increasing functions so that indifference may only result from the stated agent preferences, and not from the assigned cardinal utility values. A profile of anointments is denoted $\mathbf{x} = (x_1, \dots, x_n)$. We consider nonempty subsets $\mathbf{X} \subseteq \mathbf{I}$ of anointment profiles, where $\mathbf{I}$ is the set of all componentwise increasing functions from $[L]$ to $\Re^n$. A typical choice for $\mathbf{X}$ is all possible anointments, that is, $\mathbf{X} = \mathbf{I}$. Proper subsets $\mathbf{X}$ of $\mathbf{I}$ may be used to model, for instance, situations where the cardinal utilities of certain agents are restricted to certain subsets of possible values.

Given a social choice function f, label arcs in $G_{i,\mathbf{t}_{-i}}$ with *weight formula*

$$w(u, t) = x_i(t(f(t, \mathbf{t}_{-i}))) - x_i(t(f(u, \mathbf{t}_{-i})))$$

based on anointment x_i for all $u \neq t \in T_i$. Since anointments are strictly increasing functions, the sign of $w(s, t)$ is precisely the sign of arc (s, t) in the sense defined above, irrespective of the anointment, e.g., arc (s, t) is positive if and only if $w(s, t) > 0$ for every strictly increasing anointment x_i. Unlike arcs, the sign of *cycles* may depend on the choice of anointment. The *weight* of cycle $C = (t_{(0)}, t_{(1)}, \dots, t_{(k)}, t_{(0)})$ in $G_{i,\mathbf{t}_{-i}}$ is $x_i(C) = \sum_{j=0}^{k} w(t_{(j)}, t_{(j+1)})$ where $t_{(k+1)} = t_{(0)}$. Our new notions of non-manipulability are defined in terms of these cycle weights.

Let $G_{i,\mathbf{t}_{-i}}(x_i)$ denote the anointed version of $G_{i,\mathbf{t}_{-i}}$ for a given anointment x_i, i.e., with all its arc weight formulas evaluated at the given x_i. We say a social choice function f is *implementable* with respect to a set $\mathbf{X}$ of anointment profiles if no anointed type graph $G_{i,\mathbf{t}_{-i}}(x_i)$ contains a negative weight cycle for any anointment profile $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbf{X}$. When $\mathbf{X} = \mathbf{I}$ we say f is *fully implementable* if these conditions hold.

Put in other words, we call a cycle C in $G_{i,\mathbf{t}_{-i}}$ *structurally nonnegative* with respect to $\mathbf{X}$ if $x_i(C) \geq 0$ for all $\mathbf{x} \in \mathbf{X}$. Then f is implementable with respect to $\mathbf{X}$ if and only if all the cycles in all the type graphs $G_{i,\mathbf{t}_{-i}}$ are structurally nonnegative with respect to $\mathbf{X}$.

When a social choice function is not fully implementable but is nonetheless implementable with respect to a nonempty strict subset $\mathbf{X}$ of $\mathbf{I}$ we call the social choice function *partially implementable*. Note that every strategy-proof social choice function is fully implementable, since no negative weight cycles can arise when there are no negative arcs.

These definitions are motivated by a result due to Rochet which characterizes implementability (with utility transfers) of allocation rules in mechanism design in terms of negative weight cycles in graphs. We can state this result in our context as follows. An anointment x_i turns a type $t \in T_i$ into a *cardinal type* $x_i(t) = (x_i(t(a)) : a \in A) \in \mathbb{R}^A$. Thus we can map the type space T_i into the cardinal type space $x_i(T_i) = \{x_i(t) : t \in T\}$. Similarly, with an anointment profile $\mathbf{x} = (x_1, \ldots, x_n)$ we map the set of type profiles $\mathbf{T}$ to the set $\mathbf{x}(\mathbf{T}) = \{\mathbf{x}(\mathbf{t}) : \mathbf{t} \in \mathbf{T}\} \subseteq \prod_{i=1}^n x_i(T_i)$ of cardinal type profiles. Anointment profile $\mathbf{x}$ also turns a given social choice function f into a *cardinal allocation rule* $f_{\mathbf{x}} : \mathbf{x}(\mathbf{T}) \to A$. In the context and terminology of mechanism design, the allocation rule $f_{\mathbf{x}}$ is *implementable with utility transfers* (under quasilinear utilities and in dominant strategies) if there exists a *payment rule* $p : \mathbf{x}(\mathbf{T}) \to \mathbb{R}^n$ such that for every agent i, every cardinal type profile $(\theta_i, \boldsymbol{\theta}_{-i}) \in \mathbf{x}(\mathbf{T})$ and every alternate cardinal type declaration $\sigma_i \in x_i(T_i)$,

$$\theta_i(f_{\mathbf{x}}(\theta_i, \boldsymbol{\theta}_{-i})) - p_i(\theta_i, \boldsymbol{\theta}_{-i}) \geq \theta_i(f_{\mathbf{x}}(\sigma_i, \boldsymbol{\theta}_{-i})) - p_i(\sigma_i, \boldsymbol{\theta}_{-i}). \tag{1}$$

That is, for each agent, the *net utility* for stating her true type is no less than the net utility of declaring any alternate type, regardless of the stated types of the other agents. Then we have, restated in our context:

Theorem 1 (Rochet) *Let f be a social choice function and $\mathbf{x} = (x_1, \ldots, x_n)$ an anointment profile. The associated cardinal allocation rule $f_{\mathbf{x}}$ is implementable with utility transfers if and only if no anointed type graph $G_{i,\mathbf{t}_{-i}}(x_i)$ has a negative weight cycle.*

Thus our definition of implementability of a social choice function $f : \mathbf{T} \to A$ with respect to a set of anointment profiles $\mathbf{X}$ can be interpreted as requiring that for every $\mathbf{x} \in \mathbf{X}$ the associated cardinal allocation rule $f_{\mathbf{x}} : \mathbf{x}(\mathbf{t}) \to A$ is implementable with utility transfers.

A social choice function which is fully implementable with respect to a specified set $\mathbf{X}$ of anointments has the following appeal: although it may not be strategy-proof, for every anointment in $\mathbf{X}$ there exist utility transfers which induce truth-telling. The following example shows that full implementability is indeed distinct from strategy-proofness and Pareto optimality.

Example 1 Consider, for simplicity, a single agent with totally ordered preferences on three alternatives, i.e., $\mathbf{T} = T_1 = \{(1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1)\}$, and the social choice function f that maps all types in $\mathbf{T}$ to a except for $f(1,3,2) = b$. Noting that $\mathbf{t}_{-1}$ is void, the resulting type graph $G_{1,\mathbf{t}_{-1}}$ is illustrated in Figure 1. Observe that f is not Pareto optimal since, for example, $f(2,3,1) = a \neq b$, while b is the most preferred alternative for this type. Moreover f is not strategy-proof since for $u = (1,3,2)$ and $t = (1,2,3)$ we have $t(f(t,\mathbf{t}_{-1})) = t(a) = 1 < 2 = t(b) = t(f(u,\mathbf{t}_{-1}))$ (this is shown by the negative arc $((1,3,2),(1,2,3))$ in Figure 1).

Nonetheless, f is fully implementable. We argue that in this graph all cycles are structurally nonnegative. First observe the only negative arcs in the graph are $((1,3,2),(1,2,3))$ and $((1,3,2),(2,3,1))$, for these three nodes are the only ones where $t(b) > t(a)$, and thus the

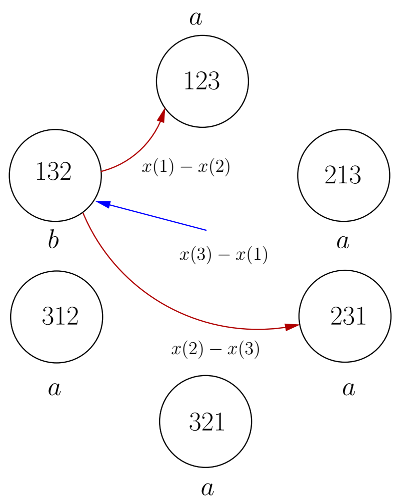

Fig. 1: Type graph for Example 1.

only ones where misrepresenting the truth might be beneficial; their weight is $x(1) - x(2)$ and $x(2) - x(3)$, respectively. Since a cycle can leave node $(1, 3, 2)$ at most once, a negative cycle must contain exactly one of these two negative arcs. Such a cycle must also include an incoming arc into node $(1, 3, 2)$, and all such arcs have weight $x(3) - x(1) > \max\{x(1) - x(2), x(2) - x(3)\}$. Thus such a cycle is actually structurally nonnegative, a contradiction. Therefore all cycles are structurally nonnegative, as claimed, i.e., f is fully implementable. □

Our next result shows that relaxing strategy-proofness to partial implementability indeed reduces the applicability of impossibility results such as Gibbard and Satterthwaite's:

Theorem 2 *For every setting with at least three alternatives, at least two agents and strict preference orderings, there exists a partially implementable social choice function f which is Pareto optimal and non-dictatorial.*

Proof Define the following social choice function. Designate one agent (agent 1) as the quasi-dictator. The quasi-dictator gets her top choice except when her top choice is the "forbidden fruit" alternative a. When the quasi-dictator's top choice is a, the best choice of the lowest-index dissenting agent is chosen. Moreover, alternative a is chosen when *all* agents declare a as their top choice. We can write the social choice function formally as

$$f(\mathbf{t}) = \begin{cases} a & t_i(a) = |A| \text{ for all } i \in \{1, \ldots, n\} \\ b & b \neq a \text{ and } t_1(b) = |A| \\ c & c \neq a \text{ and } t_i(a) = |A| \text{ for } i \in \{1, \ldots, k-1\} \\ & \qquad\qquad \text{but } t_k(c) = |A| \text{ for some } k \in \{2, \ldots, n\}\,. \end{cases}$$

Note that f is Pareto optimal. Indeed, for every type profile $\mathbf{t}$, $f(\mathbf{t})$ is some agent's top choice. This means no alternative is Pareto dominated by any other. Also, it is clear that f is not dictatorial.

We next show that f is partially implementable. We will show that there exists a type graph for the quasi-dictator which has a cycle which is undetermined in sign, and that no type graph (including those of other agents) has structurally negative cycles.

First consider the quasi-dictator's type graph $G_{1,\mathbf{t}_{-1}}$. When $\mathbf{t}_{-1}$ has $t_i(a) = |A|$ for all $i \in \{2, \dots, n\}$ then the quasi-dictator is always allocated her top choice. Thus, in this case, every anointed type graph has only nonnegative length arcs, meaning all cycles are structurally nonnegative.

Thus we may assume that in rival type profile $\mathbf{t}_{-1}$ there exists a rival agent whose most preferred alternative is not a. Similar to the above, all arcs that do not involve nodes t where $t(a) = |A|$ are nonnegative. Now, let t be a node in $G_{1,\mathbf{t}_{-1}}$ where $t(a) = |A|$ and $f(t, \mathbf{t}_{-1}) = c \neq a$. All outgoing arcs from t are nonnegative. Indeed, if the head u of such an arc is a node allocated to c, then the arc has weight zero. This includes the case where $u(a) = |A|$. Otherwise, if u is a node such that $f(u, \mathbf{t}_{-1}) = b \neq c$ then $u(b) = |A|$ and hence $w(t, u) = x(|A|) - x(u(c)) > 0$. As for *incoming* arcs, the sign depends on the allocation of the tail node v. By our assumptions, $f(t, \mathbf{t}_{-1}) = c \neq a$. Let v be a node where $v(b) = |A|$ with $b \neq a$ and $t(b) > t(c)$. This implies $f(v, \mathbf{t}_{-1}) = b$ and so $w(v, t) = x(t(c)) - x(t(b)) < 0$. However if $t(b) < t(c)$, then $w(v, t) > 0$. This implies that every negative arc in $G_{1,\mathbf{t}_{-1}}(x_1)$ is of the form (v, t) where $t(a) = |A|$, $v(b) = |A|$ and $t(b) > t(c)$.

We now claim that $G_{1,\mathbf{t}_{-1}}$ has no structurally negative cycles. Any cycle C with a chance of being structurally negative must contain a node with the properties of t. In this case we need at least one incoming arc and one outgoing arc from t in the cycle with $t(a) = |A|$. All outgoing arcs (t, u) are nonnegative and the only negative incoming arcs (v, t) are those where $v(b) = |A|$ with $b \neq a$ and $t(b) > t(c)$.

Now, $w(v, t) = x(t(c)) - x(t(b)) \geq x(1) - x(|A| - 1)$ since x is strictly increasing and since $t(a) = |A|$, $t(b) \leq |A| - 1$. Note also that $w(t, u)$ is either 0 if $f(u, \mathbf{t}_{-1}) = c$ or $w(t, u) = x(|A|) - x(u(c))$ otherwise. In order to complete the cycle C, there will be an arc from a node t' allocated to c to a node u' allocated to $d \neq c, a$. This arc has weight $w(t', u') = x(|A|) - x(u'(c))$ where we use the fact that since $f(u', \mathbf{t}_{-1}) = d$ and $d \neq c$ we must have $u'(b) = |A|$. This means for every node t in cycle C we can pair the weight $w(v, t)$ with weight $w(t', u')$. Now observe

$$\begin{aligned} w(v,t) + w(t',u') &= x(t(c)) - x(t(b)) + x(|A|) - x(u'(c)) \\ &\geq x(1) - x(|A|-1) + x(|A|) - x(|A|-1) \\ &= x(|A|) + x(1) - 2x(|A|-1). \end{aligned} \tag{2}$$

Since $u'(d) = |A|$ we know $u'(c) \leq |A| - 1$. Thus, we can conclude that for anointments x where $x(|A|) + x(1) - 2x(|A|-1) \geq 0$ we must have $w(C) \geq 0$. That is, there are no structurally negative cycles.

We now show that there exists a cycle in $G_{1,\mathbf{t}_{-1}}$ which is undetermined in sign for some choice of $\mathbf{t}_{-1}$. Consider a node t in $G_{1,\mathbf{t}_{-1}}$ where $t(a) = |A|$ and $f(t, \mathbf{t}_{-1}) = c \neq a$. Let u be a node where $u(b) = |A|$ with $b \neq a$ and $t(b) > t(c)$. Suppose also that t and u are chosen so that $t(c) = 1$ and $t(b) = u(c) = |A| - 1$. Observe that cycle $C = (u, t, u)$ is undetermined in sign. Indeed $w(C) = x(t(c)) - x(t(b)) + x(u(b)) - x(u(c)) = x(|A|) + x(1) - 2x(|A| - 1)$ which can be positive or negative depending on the choice of anointment x.

It remains to show that the type graphs of the remaining agents have no structurally negative cycles. Consider agent k where $1 < k \leq n$. There are three cases to consider. First, if there exists an i where $1 \leq i < k$ with $t_i(b) = |A|$ with $b \neq a$ then $G_{k,t_{-k}}$ has no negative arcs. This follows since all nodes in $G_{k,t_{-k}}$ are allocated to b and so all arcs have zero length. Second, if all rival agents most prefer a, then agent k gets his top choice no matter his type choice. Again, this means there are no negative arcs in this type graph.

Finally, the remaining case is where $t_i(a) = |A|$ for $i = 1, \dots, k-1$ and there exists an $\ell \in \{k+1, \dots, n\}$ such that $t_\ell(c) = |A|$ where $c \neq a$. In this setting, agent k is a residual quasi-dictator, whenever his most preferred alternative is different from a, he is allocated that alternative by declaring his true preferences. However, a is a forbidden fruit alternative just as it was for the quasi-dictator. This, analysis nearly identical to that of the quasi-dictator will show that there are no structurally negative cycles in the type graphs associated with this case. This shows f is partially implementable. □

3 Scoring rules

The approach of assigning cardinal values to preference levels is reminiscent of scoring rules. In scoring rules, agents provide strict preference orderings (that is, $T_i = [|A|]!$ for all $i = 1, \ldots, n$) and each preference level is "scored" according to a score vector $s \in \mathbb{R}^{|A|}$ which is common across all agents. The score vector is increasing with preference level. The alternative chosen by the social choice rule is the one with largest total score summed across all agents, using a specified tie-breaking rule. In other words, for every $\mathbf{t} = (t_1, \ldots, t_n) \in \mathbf{T}$, $f(\mathbf{t})$ is chosen in $\arg\max_{a \in A} \sum_{i=1}^{n} s(t_i(a))$. A classic example is the voting system proposed by Borda where $s_\ell = \ell$ for all $\ell \in [L]$ (see, e.g., Gaertner (2009)). Scoring rules leave open the possibility of ties in total score across alternatives, hence the need for a systematic way to break ties.

Note that we are using the terminology *social choice rule* as opposed to social choice function when referring to scoring rules. This is because scoring rules are typically stated without specifying the number of agents nor the number of alternatives. Fixing the number of actions and the number of agents defines a *social choice setting* which instantiates a social choice rule as a social choice function.

It is well known that every scoring rule is Pareto optimal and, with two or more agents, not dictatorial. On the other hand, it is also well known that scoring rules with three or more alternatives are not strategy-proof (see, e.g., Gaertner (2009)). In contrast we have:

Theorem 3 *Every scoring rule is implementable with respect to the anointment profile $\mathbf{x} = \mathbf{s}$ whereby each agent i uses the score vector s as anointment, i.e., where every $x_i = s$.*

Proof Consider a scoring rule defined by a score vector $s \in \Re^L$, and an instantiation f with a set A of alternatives (so $|A| = L$) and n agents. For any agent i and rival type profile $\mathbf{t}_{-i}$, consider a cycle $C = (t^0, t^1, t^2, \ldots, t^p)$, where $t^p = t^0$, in the anointed type graph $G_{i,\mathbf{t}_{-i}}(\mathbf{s})$ defined by using the score vector s as anointment for each agent. For $j = 1, \ldots, p$, let $a_j = f(t^j, \mathbf{t}_{-i})$ be the chosen outcome when agent i states t^j as his preference ordering. The weight of this cycle is $w(C) = \sum_{j=1}^{p} \big(s(t^j(a_j)) - s(t^j(a_{j-1}))\big)$. Let $S_{-i}(a) = \sum_{k \neq i} s(t_k(a))$ denote the total score of alternative $a \in A$ resulting from the rival type profile $\mathbf{t}_{-i}$. For every $j = 1, \ldots, p$, the total score of the chosen alternative a_j according to profile $\mathbf{t}^j = (t^j, \mathbf{t}_{-i})$ must be at least that of alternative a_{j-1} according to the same profile $\mathbf{t}^j$, i.e., $0 \leq \big(S_{-i}(a_j) + s(t^j(a_j))\big) - \big(S_{-i}(a_{j-1}) + s(t^j(a_{j-1}))\big)$. Adding these inequalities, all total rival scores $S_{-i}(a_j)$ cancel out and we get $0 \leq \sum_{j=1}^{p} \big(s(t^j(a_j)) - s(t^j(a_{j-1}))\big) = w(C)$. This shows that every cycle in every score-anointed type graph $G_{i,\mathbf{t}_{-i}}(s)$ has nonnegative weight. By Rochet's Theorem, this implies that the instantiation f of the scoring rule defined by the score vector s is implementable.

Several remarks about this theorem and its proof are in order. First, anointments are thus related to scores, beyond the mere fact that both are increasing functions, in particular scores may be used as a special case of anointments. Next, we may interpret implementability in this case by considering scores as a currency that affects agents' utilities and can be transferred through payment rules.

Scoring rules leave open the possibility of ties in total score across alternatives, hence the need for a systematic way to break ties. Call a tie-breaking rule *consistent* if it breaks ties according to a specified total order, say, $a_1 \succ a_2 \succ \cdots \succ a_{|A|}$ of the alternative set A, i.e., $f(\mathbf{t}) = \max_{\succ}(\arg\max_{a\in A} \sum_{i=1}^{n} s(t_i(a)))$ for all $\mathbf{t} \in \mathbf{T}$. Thus, for example, $f(\mathbf{t}) = a_1$ whenever a_1 is an alternative with highest total score; $f(\mathbf{t}) = a_2$ whenever a_2 is an alternative with highest total score and a_1 is not; etc. Note that the proof of Theorem 3 applies irrespective of how ties in total score are broken, i.e, is independent of a tie-breaking rule.

The following result fully characterizes what can happen in the case of three alternatives in terms of implementability.

Theorem 4 *A 3-outcome scoring rule with a consistent tie-breaking rule is implementable with respect to an anointment vector* $\mathbf{x}$ *if and only if its score vector* $\mathbf{s}$ *satisfies*

$$(2\,s(2) - s(3) - s(1))\,(2\,x_i(2) - x_i(3) - x_i(1)) \geq 0 \quad \textit{for all } i \in \{1,\ldots,n\}. \tag{3}$$

In particular, the Borda scoring rule with three outcomes is fully implementable.

Proof ($\Rightarrow$) Proof by contrapositive. Fix the agent, say agent 1 and drop the subscripts on the x's. Also assume (without loss) the tie-breaking order $a \succ b \succ c$. Suppose

$$(2\,s(2) - s(3) - s(1))\,(2\,x_i(2) - x_i(3) - x_i(1)) < 0. \tag{4}$$

This can happen in one of two ways.

Case A: Condition (4) holds since

$$2s(2) - s(3) - s(1) < 0 \tag{5}$$

and

$$2x(2) - x(3) - x(1) > 0. \tag{6}$$

or

Case B: Condition (4) holds since

$$2s(2) - s(3) - s(1) > 0 \tag{7}$$

and

$$2x(2) - x(3) - x(1) < 0. \tag{8}$$

Consider Case A. We construct a social setting and a type graph appropriate to that setting which has a negative arc under any anointment satisfying (6).

Consider the rival profile $t_{-1} = ((3,2,1),(3,2,1),(2,3,1),(2,1,3),(2,1,3))$ in the social choice setting with five rivals. It is straightforward to verify that these rival profiles, along with (5) give rise to the type graph $G_{1,t_{-1}}$ shown in Figure 2. Note that cycle $C = ((1,2,3),(2,3,1),(1,2,3))$ has weight formula $w(C) = x(1) + x(3) - 2x(2)$ in this type graph. Thus, this cycle has negative weight for any anointment satisfying (6).

A similar construction arises in Case B. The rival profile

$$t_{-1} = ((3,2,1),(3,2,1),(3,2,1),(2,3,1),(2,3,1))$$

in the social choice setting with five rivals, along with (7), gives rise to the type graph $G_{1,t_{-1}}$ shown in Figure 3. Note that cycle $C = ((1,2,3),(2,3,1),(1,2,3))$ has weight formula $w(C) = 2x(2) - x(1) - x(3)$ in this type graph. Thus, this cycle has negative weight for any anointment satisfying (8).

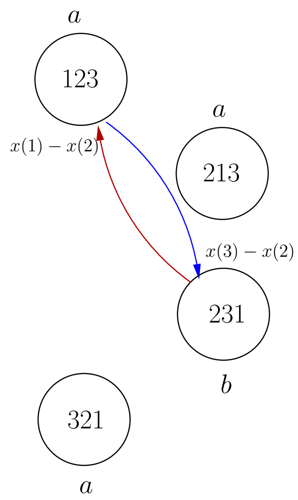

Fig. 2: Type graph in Case A with a negative cycle.

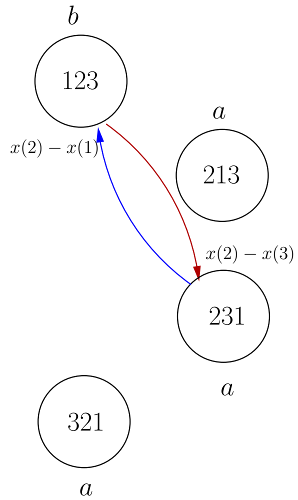

Fig. 3: Type graph in Case B with a negative cycle.

Thus in both cases we produce negative length cycles under condition (4), thus establishing the forward direction of the proof.

($\Leftarrow$) Under the stated assumptions let, without loss of generality, the score vector

$$s = (s(1), s(2), s(3)) = (0, \sigma, 1) \text{ with } 0 < \sigma < 1.$$

Again, fix an agent, say agent 1 and drop the subscripts on the x's. Define vector

$$S_{-1} = (S_{-1}(a), S_{-1}(b), S_{-1}(c))$$

where $S_{-1}(a) = \sum_{i \neq 1} s_{t_i(a)}$ and $S_{-1}(b)$ and $S_{-1}(c)$ are defined analogously. We call S_{-1} the vector of rival total scores. Note that every type graph $G_{1,\mathbf{t}_{-1}}$ where $\mathbf{t}_{-1}$ gives rise to the rivals' score vector S_{-1} is identical. Therefore it suffices to search for negative cycles in one type graph for every rival total score vector. Let $G_{1,S_{-1}}$ denote the type graph which arises from rivals' score vector is S_{-1}. We show that when (3) holds there are no structurally negative cycles in any type graph $G_{1,S_{-1}}$ where S_{-1} ranges over all possible rival scores (and thus all social choice settings with three outcomes). The same argument would hold for each agent, thus establishing the result.

We start by examining what structure is implied in the graph $G_{1,S_{-1}}$ if negative arcs of certain weight formulas appear. We need not consider those type graphs with no negative arcs, since no negative cycles can exist in such graphs. There are three types of negative arcs weights which could appear: i) $x(1) - x(3)$, ii) $x(1) - x(2)$ and iii) $x(2) - x(3)$.

First, we claim there are no arcs with weight $x(1) - x(3)$ in any type graph. Suppose, by way of contradiction, that there exists an arc (u, t) in some type graph $G_{1,S_{-1}}$ with arc weight formula $x(1) - x(3)$. Since the weight of this arc is nonzero, we must have $f(u) \neq f(t)$.[2] Assume, without loss, that $f(t) = a$ and $f(u) = b$. This immediately implies that $t = (3, 2, 1)$. For f to choose a under type declaration t it must mean that

$$S_{-1}(a) + 0 \geq S_{-1}(b) + 1 \tag{9}$$

since $s(t(a)) = 0$ and $s(t(b)) = 1$. Note there is a possibility that this inequality is strict if $a \prec b$ in the tiebreaker order. We now derive a contradiction of the fact $f(u) = b$. Indeed, for a to be allocated to u we must have

$$S_{-1}(b) + s(u(b)) \geq S_{-1}(a) + s(t(a)) \tag{10}$$

again with the possibility of a strict inequality instead if $a \succ b$. Note since we cannot have both $a \succ b$ and $b \succ a$, this implies that at least one of the inequalities in (9) and (10) is strict.

However, now it follows that $S_{-1}(b) + s(u(b)) \geq S_{-1}(a) + s(u(a)) \geq S_{-1}(a) + 0 \geq S_{-1}(b) + 1$ with at least one of the inequalities strict. This, however, is a contradiction since $s(u(b)) \in \{0, \sigma, 1\}$. We conclude that no arc of weight $x(1) - x(3)$ exists.

The remaining possibilities can be classified into two cases. The type graphs with an arc of weight $x(1) - x(2)$ are Case 1 (C1). Graphs with only arcs of weight $x(2) - x(3)$ are in Case 2 (C2).

Consider a type graph in (C1). It possesses an arc (u, v) of weight $x(1) - x(2)$. Without loss of generality, take $f(t) = a$ and $f(u) = b$. This means that $t(a) = 1$ and $t(b) = 2$ and so $t = (1, 2, 3)$. Now, $f(t) = a$ if only if

$$S_{-1}(a) + 0 \geq S_{-1}(b) + \sigma \tag{11}$$

$$S_{-1}(a) + 0 \geq S_{-1}(c) + 1 \tag{12}$$

where the weak inequalities may become strict depending on the tiebreaker order.

Note that (12) implies $f(v) \neq c$ for all types $v \in T_1$. This is immediate if $a \succ c$ since $S_{-1}(a) + 0$ is the smallest possible total score for alternative a and $S_{-1}(c) + 1$ is the largest possible score for alternative c.

Thus, f only assigns alternatives a and b. We turn to characterizing which nodes can be assigned alternative b in the type graph and be consistent with the scoring rule and tiebreak ordering. Note that for all $v \in G_{1,S_{-1}}$, $f(v) = b$ if and only if

$$S_{-1}(b) + s(v(b)) \geq S_{-1}(a) + s(v(a)),$$

[2] Note the more precise notation here is $f(u, \mathbf{t}_{-1}) \neq f(t, \mathbf{t}_{-1})$, but we are suppressing the additional appending of $\mathbf{t}_{-1}$ for brevity.

again where the inequality is strict when $b \prec a$. Combining this with (11) yields:

$$\sigma \leq S_{-1}(a) - S_{-1}(b) \leq s(v(b)) - s(v(a)) \tag{13}$$

where at least one of these inequalities is strict, irrespective of the tiebreak ordering. Note that this expression forbids certain values for $s(v(b))$ and $s(v(a))$. For instance, we cannot have $s(v(b)) = \sigma$ and $s(v(a)) = 0$ since it violates the inequalities (recall at least one must be a strict inequality). In fact, only two possibilities remain: $s(v(b)) = 1$ and $s(v(a)) = 0$, and $s(v(b)) = 1$ and $s(v(a)) = \sigma$ only if $\sigma < 1/2$. This indicates that the only possible nodes that can be assigned allocation b are $(1, 3, 2)$ and $(2, 3, 1)$ respectively. Moreover, observe that if $f(2, 3, 1) = b$ we require (in addition to $\sigma < 1/2$) that

$$\sigma \leq S_{-1}(a) - S_{-1}(b) \leq 1 - \sigma, \tag{14}$$

which immediately implies that $\sigma < S_{-1}(a) - S_{-1}(b) < 1$ and so $f(1, 3, 2) = b$ also.

Since we have assumed that a negative arc exists, and so at least one node is allocated to b, there are only two type graphs possible. We already considered the first in Example 1, the type graph is Figure 1, where we established all cycles are structurally nonnegative. The second we have also seen, Figure 2, which has $f(1, 3, 2) = f(2, 3, 1) = b$ and $f(t) = a$ otherwise. This arises only if $\sigma < 1/2$.

Recall, in this type graph there exists the 2-cycle $C = ((1, 2, 3), (2, 3, 1), (1, 2, 3))$, shown in Figure 2. The weight of the cycle is $w(C) = x(1) + x(3) - 2x(2)$. However, since $\sigma < 1/2$, by (3) we must have $x(1) + x(3) - 2x(2) > 0$. Thus, C is a nonnegative cycle. Hence the type graph in Figure 2 has no negative cycles under our hypothesis. This completes the treatment of (C1).

Consider case (C2). We assume the type graphs under consideration contain an arc (u, t) of weight $x(2) - x(3)$. Without loss of generality, take $f(t) = a$ and $f(u) = b$. This means that $t(a) = 2$ and $t(b) = 3$ and so $t = (2, 3, 1)$. Now, $f(t) = a$ if only if

$$S_{-1}(a) + \sigma \geq S_{-1}(b) + 1 \tag{15}$$

$$S_{-1}(a) + \sigma \geq S_{-1}(c) + 0, \tag{16}$$

where the inequalities may become strict depending on the tiebreaker order.

Unlike in (C1) we cannot argue that all nodes are either allocated to a or b. However, we proceed with an analysis similar to (C1) and ask which nodes can be allocated to b. We have supposed, again without loss, that at least one node is allocated to b. Now, for all nodes v, $f(v) = b$ if and only if $S_{-1}(b)+s(v(b)) \geq S_{-1}(a)+s(v(a))$ and $S_{-1}(b)+s(v(b)) \geq S_{-1}(c)+s(v(c))$. Combining this with (15) yields: $1-\sigma \leq S_{-1}(a)-S_{-1}(b) \leq s(v(b))-s(v(a))$ where at least one of these inequalities is strict, irrespective of the tiebreak ordering. Only two combinations of values for $s(v(b))$ and $s(v(a))$ are possible: $s(v(b)) = 1$ and $s(v(a)) = 0$, and $s(v(b)) = \sigma$ and $s(v(a)) = 0$ only if $\sigma > 1/2$. This indicates that the only possible nodes that can be assigned allocation b are $(1, 3, 2)$ and $(1, 2, 3)$. Observe that if $f(1, 2, 3) = b$, then we immediately know that $f(1, 3, 2) = b$ since the total score for b can only increase and the total scores for the other alternatives can only decrease (in the case of c) or be left unchanged (as in the case of a) as compared to node $(1, 2, 3)$. Since we have assumed that there is always at least one node allocated to b we can therefore reduce to two possible sets of nodes allocated to b: $\{(1, 3, 2)\}$ and $\{(1, 2, 3), (1, 3, 2)\}$. We treat these as two subcases:

(C2.1) $\quad f^{-1}(b) = \{(1, 3, 2)\}$,

(C2.2) $\quad f^{-1}(b) = \{(1, 2, 3), (1, 3, 2)\}$ only if $\sigma > 1/2$

In both cases we are left to assign the remaining nodes to either a or c. However, the allocation of some of these nodes is implied. Above we assumed, without loss, that $f(2, 3, 1) = a$. This

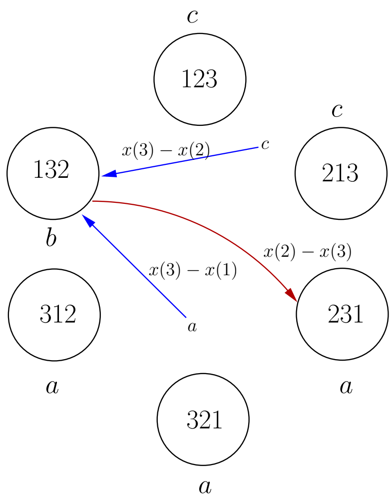

Fig. 4: Type graph for Case (C2.1) with $f(2,1,3) = c$.

implies $f(3,2,1) = a$ since in this new node the total score for a has only increased, whereas the score for c has decreased. This means that when a has the highest score in $(2,3,1)$ it must have the highest score in $(3,2,1)$.

A more subtle implication is that we always have $f(3,1,2) = a$. Indeed for $f(3,1,2) = c$ we would need both

$$S_{-1}(c) + \sigma \geq S_{-1}(a) + 1 \tag{17}$$

$$S_{-1}(c) + \sigma \geq S_{-1}(b) + 0 \tag{18}$$

However, these conditions give rise to a contradiction. Indeed, putting together (16) and (17) yields $1 - \sigma \leq S_{-1}(c) - S_{-1}(a) \leq \sigma$ where at least one of the inequalities in strict. This can only be true if $\sigma > 1/2$. Now, since in both (C2.1) and (C2.2) has $f(1,3,2) = b$ we get the implication $S_{-1}(b) + 1 \geq S_{-1}(c) + \sigma$ which when combined with (18) yields $\sigma \leq S_{-1}(c) - S_{-1}(a) \leq 1 - \sigma$ which can only happen when $\sigma < 1/2$ since at least of these inequalities is strict. Thus for both (17) and (18) to hold would require $\sigma > 1/2$ and $\sigma < 1/2$, a contradiction. Thus, we conclude that $f(3,1,2) = a$.

Thus, in subcase (C2.1) we are left only to allocate alternatives a and c to nodes $(1,2,3)$ and $(2,1,3)$. Note that we can immediately conclude that $f(1,2,3) = c$. The reason is simple: if $f(1,2,3) = a$, this would introduce a weight of $x(1) - x(2)$ onto the arc $((1,3,2),(1,2,3))$. However, from the second claim we argued that if there is an arc of weight $x(1) - x(2)$ then all negative arcs must have this weight. This is contradicted by the fact we have arc $((1,3,2),(2,3,1))$ of length $x(2) - x(3)$. Thus, $f(1,2,3) = c$.

We are left to allocate an alternative to node $(2,1,3)$. Suppose $f(2,1,3) = c$. The resulting type graph is Figure 4. Analysis reveals that the only negative arc is $((1,3,2),(2,3,1))$ with weight $x(2) - x(3)$. For this arc to appear in a cycle it would need to include an incoming arc into b. All such arcs have weight at least $x(3) - x(2)$. Hence the net contribution to the weight of cycle by entering $(1,3,2)$ and leaving along the negative arc $((1,3,2),(2,3,1))$ is nonnegative. This implies that all cycles are structurally nonnegative.

Suppose instead $f(2,1,3) = a$. The resulting type graph is Figure 5. Here there are two negative arcs $((1,3,2),(2,3,1))$ and $((1,2,3),(2,1,3))$ both with weight $x(2) - x(3)$. Any negative

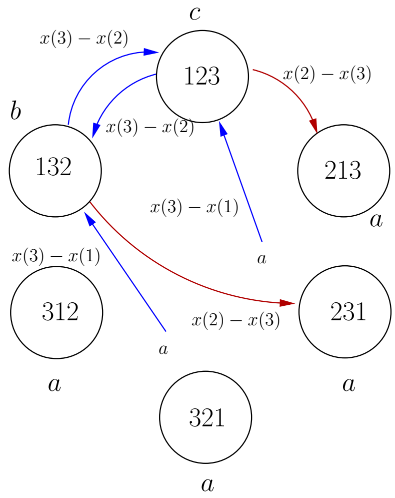

Fig. 5: Type graph for Case (C2.1) with $f(2,1,3)=a$.

cycle using arc $((1,3,2),(2,3,1))$ must enter node $(1,3,2)$. However, all arcs entering this node have positive weight of at least $x(2)-x(3)$. Thus, again, the net to cycle weight of using arc $((1,3,2),(2,3,1))$ is nonnegative. Similar reasoning holds for using arc $((1,2,3),(2,1,3))$ in a cycle noting the fact that all arcs entering node $(1,2,3)$ have positive weight.

Returning to case (C2.2) we are left only to allocate an alternative to node $(2,1,3)$. We claim that we must have $f(2,1,3)=a$. Suppose $f(2,1,3)=c$. Since $f(1,2,3)=b$, $f(2,3,1)=a$ and $f(2,1,3)=c$ we get the system of inequalities:

$$S_{-1}(b)+\sigma \geq S_{-1}(c)+1$$
$$S_{-1}(a)+\sigma \geq S_{-1}(b)+1$$
$$S_{-1}(c)+1 \geq S_{-1}(a)+\sigma.$$

Summing these inequalities together simplifies to $\sigma \geq 1$, a contradiction. This leaves only $f(2,1,3)=a$, which have seen before in Figure 3. All cycles are structurally nonnegative except for $C=((1,2,3),(2,3,1),(1,2,3))$ which has weight formula $w(C)=2x(2)-x(1)-x(3)$. However since this type graph can only appear when $\sigma>1/2$, by (3) it follows that $2x(2)-x(1)-x(3)\geq 0$. Hence, C is nonnegative under our hypothesis.

Having undertaken a complete analysis of all possible cases we have shown that no negative cycles exist in any type graphs when anointments satisfy (3). This establishes the result. □

The previous result implies that full implementability and Pareto optimality together need not imply dictatorship. Indeed, the Borda rule is not a dictatorship but it is both Pareto optimal and, for 3 outcomes, fully implementable. On the other hand, once we consider four or more alternatives, no scoring rule is fully implementable.

Theorem 5 *Every scoring rule with four or more alternatives and a consistent tie-breaking rule has a two-player instantiation which is not fully implementable.*

Proof Consider a scoring rule with alternatives $A=\{a_1,\ldots,a_L\}$ where $|A|=L\geq 4$, score vector $\mathbf{s}$, and tie-breaking rule $a_1 \succ a_2 \succ \cdots \succ a_L$. We distinguish two cases for $\mathbf{s}$, and in each case we exhibit a two-player instantiation, a 2-cycle C and an anointment for which C has negative weight.[3]

(i) Assume $s(L-1)+s(L-2) \geq s(L)+s(1)$. Consider a 2-agent instantiation and the type graph $G_{1,\mathbf{t}_{-1}}$ where agent 2's profile $\mathbf{t}_{-1} = t$ has components $t(a_1) = L$, $t(a_2) = L-1$, and $t(a_j) = j-2$ for $j = 3,\ldots,L$. Consider the 2-cycle $C = (u,v,u)$ in $G_{1,\mathbf{t}_{-1}}$ where agent 1's profiles u and v have components $u(a_1) = L-1$, $u(a_2) = L$, $u(a_3) = 1$; $v(a_1) = 1$, $v(a_2) = L-1$, $v(a_3) = L$; and $u(a_j) = v(a_j) = t(a_j) = j-2$ for $j = 4,\ldots,L$. For profile (u,t) the total scores $S(a)$ of alternatives are $S(a_1) = s(L)+s(L-1) = S(a_2) > S(a_j)$ for all $j \geq 3$, so by the tie-breaking rule $f(u,t) = a_1$. For profile (v,t) the total scores of alternatives are $S(a_2) = 2\,s(L-1) > s(L-1)+s(L-2) \geq s(L)+s(1) = S(a_1) = S(a_3)$, and $S(a_j) \leq 2\,s(L-2) < S(a_2)$ for all $j \geq 3$, so $f(v,t) = a_2$. The weight of cycle C is $2\,x_1(L-1) - x_1(L) - x_1(1)$, which is negative, e.g., for anointment x_1 with components $x_1(L) = 2L$ and $x_1(l) = l$ for all $l \leq L-1$.

(ii) Else $s(L-1)+s(L-2) < s(L)+s(1)$. Consider a 2-agent instantiation and the type graph $G_{1,\mathbf{t}_{-1}}$ where agent 2's profile $\mathbf{t}_{-1} = t$ now has components $t(a_j) = L+1-j$ for all $j = 1,\ldots,L$. Consider the 2-cycle $C = (u,v,u)$ in $G_{1,\mathbf{t}_{-1}}$ where agent 1's profiles u and v now have components $u(a_1) = 1$, $u(a_2) = L-2$, $u(a_3) = L-1$, $u(a_j) = j-2$ for $j = 4,\ldots,L-1$, and $u(a_L) = L$; and $v(a_1) = L-2$, $v(a_2) = L$, $v(a_3) = L-1$; and $v(a_j) = t(a_j) = L+1-j$ for $j = 4,\ldots,L$. For profile (u,t) the total scores of alternatives are $S(a_1) = S(a_L) = s(L)+s(1) > s(L-1)+s(L-2) = S(a_2) = S(a_3) > S(a_j)$ for all $j = 4,\ldots,L-1$, so by the tie-breaking rule $f(u,t) = a_1$. For profile (v,t) the total scores of alternatives are $S(a_2) = S(L)+S(L-1) > S(a_j)$ for all $j \neq 2$, so $f(v,t) = a_2$. The weight of cycle C is $x_1(L) + x_1(1) - 2\,x_1(L-2)$, which is negative, e.g., for anointment x_1 with components $x_1(1) = 0$ and $x_1(l) = l+1$ for all $l \geq 2$. □

4 Local conditions

A recent stream of economic theory has taken up the question of when certain "local conditions" guarantee truth-telling, see, e.g., Ashlagi et al (2010); Carroll (2012) and the references therein. In particular, the recent work Carroll (2012) characterizes when local conditions suffice in a variety of ordinal and cardinal settings . The main thrust of these results concern convex type spaces and the connection between "monotonocity" (a local condition) and truth-telling. Since our type spaces $\mathbf{x}([L])$ are *discrete* and thus not convex, these results do not immediately apply to our setting. Closer to our development is Mu'alem and Schapira (2008), which considers local conditions for truth-telling with transfers in a setting with discrete domains. Below we resolve an open question posed in their paper and we generalize and unify some of their results.

The local conditions concern 2-cycles in the graphs $G_{i,\mathbf{t}_{-i}}(x_i)$ defined in Section 2. A 2-cycle is a cycle with 2 edges, i.e., (u,t,u) for some $u \neq t$. We say a social choice function f is 2-*cyclic* with respect to profile and anointment sets $\mathbf{T}$ and $\mathbf{X}$ if for every agent i, rival profile $\mathbf{t}_{-i} \in \mathbf{T}_{-i}$ and anointment $\mathbf{x} \in \mathbf{X}$, the graph $G_{i,\mathbf{t}_{-i}}(x_i)$ has no 2-cycle with negative weight. Similarly, a social choice function f is *strongly* 2-*cyclic* with respect to profile and anointment sets $\mathbf{T}$ and $\mathbf{X}$ if for every agent i, rival profile $\mathbf{t}_{-i} \in \mathbf{T}_{-i}$ and anointment $\mathbf{x} \in \mathbf{X}$, every 2-cycle between nodes which are allocated different alternatives in the graph $G_{i,\mathbf{t}_{-i}}(x_i)$ has positive weight.

Our motivating question is to identify conditions on $\mathbf{T}$ and $\mathbf{X}$ which imply that a (strongly) 2-cyclic social choice function f is implementable. To start our investigation, we show that strong 2-cyclicity does not suffice to ensure implementability.

[3] This construction extends trivially to any number $n = 2 + mL = (2 \mod L)$ of players, by simply adding m groups of L players each, where in each group the j-th player's profile (for $j = 1,\ldots,L$) is the j-th circular permutation $(j, j+1, \ldots, L, 1, \ldots, j-1)$ of $[L]$, so all alternatives get the same total score from each player group.

Theorem 6 *When $|A| \geq 3$ and $L \geq 3$, there exist social choice functions that are strongly 2-cyclic and not fully implementable.*

Proof Consider a single agent setting with $A = \{a, b, c\}$, $L = 3$ and $T = \mathbf{T} = [L]^A$ dropping all subscripts to simplify notation. This example extends in a natural way to a multi-agent setting, by simply ignoring the other players' stated preferences.

Consider the social choice function f defined as follows:

$$f(t) = \begin{cases} c & \text{if } t = (1,1,2) \text{ or } (1,1,3); \\ b & \text{if } t = (1,j,k) \text{ for } j \in \{2,3\} \text{ and } k \in \{1,2,3\}; \\ a & \text{otherwise.} \end{cases}$$

For any strictly increasing x, there are several negative arcs in $G(x)$. First, every arc (s,t) with $s \in f^{-1}(c)$ and $t = (1,2,3)$ is a negative arc, since $w(s,t) = x(t(f(t))) - x(t(f(s))) = x(t(b)) - x(t(c)) = x(2) - x(3) < 0$. Next, every arc (s,t) with $s \in f^{-1}(c)$ and $t = (2,1,3)$ is also a negative arc, with weight $w(s,t) = x(t(a)) - x(t(c)) = x(2) - x(3) < 0$. Finally, every arc (s,t) with $s \in f^{-1}(b)$ and $t = (2,3,1)$ is also a negative arc, with weight $w(s,t) = x(t(a)) - x(t(b)) = x(2) - x(3) < 0$. It is easy to verify that every other arc has nonnegative weight. Therefore every negative arc in $G(x)$ has weight formula $x(2) - x(3)$.

A 2-cycle $C = (s,t,s)$ with negative weight must contain (at least) one of these negative weight arcs. If $s \in f^{-1}(c)$ and $t = (1,2,3)$, then $w(C) = x(s(c)) - x(s(b)) + x(2) - x(3)$, which is either $w(C) = x(2) - x(1) + x(2) - x(3) = 2x(2) - x(1) - x(3)$ or $w(C) = x(3) - x(1) + x(2) - x(3) = x(2) - x(1) > 0$. Similarly, if $s \in f^{-1}(c)$ and $t = (2,1,3)$ then $w(C) = x(s(c)) - x(s(a)) + x(2) - x(3)$, which is one of these same two values $2x(2) - x(1) - x(3)$ or $x(2) - x(1)$. Otherwise $s \in f^{-1}(c)$ and $t = (2,1,3)$ and then $w(C) = x(s(c)) - x(s(a)) + x(2) - x(3)$, which is again one of these same two values. Therefore there exists a negative 2-cycle if and only if $2x(2) - x(1) - x(3) < 0$. Furthermore every other 2-cycle $C = (s,t,s)$ with $f(s) \neq f(t)$ is structurally positive. Thus f is strongly 2-cyclic if and only if $2x(2) - x(1) - x(3) > 0$.

The 3-cycle $\tilde{C} = (r,s,t,r)$ with $r = (1,1,2)$, $s = (1,2,3)$ and $t = (2,3,1)$ has total weight $w(\tilde{C}) = (x(2) - x(3)) + (x(2) - x(3)) + (x(2) - x(1)) = 3x(2) - 2x(3) - x(1)$. Thus consider the anointment set

$$\tilde{\mathbf{X}} = \tilde{X} = \{x \in \mathbf{I} : 2x(2) - x(1) - x(3) > 0 \text{ and } 3x(2) - 2x(3) - x(1) < 0\}.$$

Note that $\tilde{x} = (0,3,5)$ is in $\tilde{\mathbf{X}}$, so $\tilde{\mathbf{X}} \neq \emptyset$. Now, f is strongly 2-cyclic relative to $\mathbf{T}$ and $\tilde{\mathbf{X}}$ (since $2x(2) - x(1) - x(3) = 1 > 0$ for all $x \in \tilde{\mathbf{X}}$) and yet it is not fully implementable (since the 3-cycle $\tilde{C}$ above has negative weight for all $x \in \tilde{\mathbf{X}}$). $\square$

The notion that, in mechanism design with transfers, corresponds exactly to strong 2-cyclicity is *strong monotonicity*, see Lavi et al (2003). Letting V_i denote the set of all possible valuations (or utility) vectors $v \in \mathbb{R}^A$ for each agent i, the set $\mathbf{V} = V_1 \times \ldots V_n$ of all possible valuation profiles is called the *domain*. In our setting with n agents, L preference levels and finite alternative set A, the set $[L]^A$ of all possible preference profiles for each agent is finite. Thus, given any anointment profile $\mathbf{x} \in \mathbf{I}$, the set of resulting valuation profiles is a finite, hence discrete, domain. Using the instance constructed in the proof of Theorem 6 and anointment $\tilde{x} = (0,3,5)$, yields

Corollary 1 *There exist instances of mechanism design with transfers, with a finite domain and an allocation rule that is strongly monotonic and not implementable.*

Note that the type sets in these instances has "gaps", i.e., there are no valuation profiles with a component equal to 1 or 2 (between the values 0 and 3) or 4 (between the values 3 and 5). Call a set $D \subseteq \mathbb{Z}^d$ of integer vectors *consecutive* if it has the form of a discrete cube $\{x \in \mathbb{Z}^d : a \leq x_j \leq b \text{ for all } j = 1, \dots, d\}$ for any $a \in \mathbb{Z} \cup \{-\infty\}$ and $b \in \mathbb{Z} \cup \{+\infty\}$ with $a < b$, and thus also the whole integer lattice $\mathbb{Z}^d$. The following theorem shows that in mechanism design with transfers and a consecutive integer type set, strong monotonicity implies implementability. This generalizes Theorem 4.4 (whole integer domains) and Proposition 4.6 (0/1-domains) in Mu'alem and Schapira (2008). In addition it positively answers an open question at the end of that paper.

Theorem 7 *In mechanism design with transfers, whenever the domain is integer and consecutive, every allocation rule that is strongly monotone is implementable.*

Proof Consider a consecutive integer domain $\mathbf{V}$ and an allocation rule f that is strongly monotone. By contradiction, assume that f not implementable, and thus there exists a negative cycle in a graph $G_{i,\mathbf{v}^{-i}}$. We fix this agent i and rival valuation profile $\mathbf{v}^{-i}$ and, as earlier, we omit the corresponding subscripts. Thus let G denote that graph, and let $C = (v^1, v^2, \dots, v^k, v^1)$ be a negative cycle in G with the minimum number k of nodes. This implies that all the $f(v^j)$, $j = 1, \dots, k$, are distinct. (Note that we slightly switched notations and we now let $v^1, v^2, \dots, v^k$ denote successive valuation vectors along this cycle for the *same* agent i.) Let $\bar{v} = \max_{g,j} v^j_g$ and $\underline{v} = \min_{g,j} v^j_g$ be the largest and smallest component value, respectively, appearing among the nodes of this cycle. Since $w(C) < 0$ we have $\underline{v} < \bar{v}$. Define the node values $p_1 = 0$ and $p_j = \sum_{h=2}^{j} w(v^{h-1}, v^h)$ for $j = 2, \dots, k$ (so each p_j is the length of the path from v^1 to v^j along cycle C, and the total weight of the cycle is $w(C) = p_k + w(v^k, v^1) < 0$). Without loss of generality, we may assume that all $p_j \leq 0$, for otherwise we would relabel the nodes in the cycle to start at v^h for some $h \in \arg\max\{p_j : j = 1, \dots, k\}$ (indeed, after this relabeling, the node values p' become $p'_j = p_j - p_h \leq 0$ for all $j \geq h + 1$, and $p'_j = p_k - p_h + w(v^k, v^1) + p_j < p_j - p_h \leq 0$ for all $j < h$). Define the valuation vector $\tilde{v} \in \mathbb{Z}^A$ with components

$$\tilde{v}_a = \begin{cases} \max\{\bar{v} + p_j, \underline{v}\} & \text{if } a = f(v^j) \text{ for some } j \in \{1, \dots, k\}; \\ \underline{v} & \text{otherwise.} \end{cases}$$

For all $a \in A$ we have $\tilde{v}_a \geq \underline{v}$ and (since all $p_j \leq 0$) $\tilde{v}_a \leq \bar{v}$. Since the integer domain $\mathbf{V}$ is consecutive this implies that $\tilde{v} \in \mathbf{V}$. We now argue that whatever the alternative $f(\tilde{v})$ allocated by f to $\tilde{v}$, we obtain a nonpositive 2-cycle $\tilde{C} = (\tilde{v}, v', \tilde{v})$ between $\tilde{v}$ and a node v' such that $f(v') \neq f(\tilde{v})$, thus contradicting the assumption that f is strongly monotone.

If $f(\tilde{v}) = f(v^j)$ for some $j \in \{1, \dots, k-1\}$ such that $\tilde{v}_{f(v^j)} = \bar{v} + p_j$, then $f(v^{j+1}) \neq f(v^j) = f(\tilde{v})$ and the 2-cycle $\tilde{C} = (\tilde{v}, v^{j+1}, \tilde{v})$ has weight

$$\begin{aligned} w(\tilde{C}) &= w(\tilde{v}, v^{j+1}) + w(v^{j+1}, \tilde{v}) = \left(v^{j+1}_{f(v^{j+1})} - v^{j+1}_{f(v^j)}\right) + (\tilde{v}_{f(v^j)} - \tilde{v}_{f(v^{j+1})}) \\ &\leq w(v^j, v^{j+1}) + ((\bar{v} + p_j) - (\bar{v} + p_{j+1})) = 0. \end{aligned}$$

If $f(\tilde{v}) = f(v^k)$ and $\tilde{v}_{f(v^k)} = \bar{v} + p_k$, then $f(v^1) \neq f(v^k) = f(\tilde{v})$ and the 2-cycle $\tilde{C} = (\tilde{v}, v^1, \tilde{v})$ has weight

$$w(\tilde{C}) = \left(v^1_{f(v^1)} - v^1_{f(v^k)}\right) + (\tilde{v}_{f(v^k)} - \tilde{v}_{f(v^1)}) = w(v^k, v^1) + p_k = w(C) < 0.$$

Otherwise, $\tilde{v}_{f(\tilde{v})} = \underline{v}$. Since $\tilde{v}_{f(v^1)} = \bar{v} < \underline{v} = \tilde{v}_{f(\tilde{v})}$ we must have $f(v^1) \neq f(\tilde{v})$. The 2-cycle $\tilde{C} = (\tilde{v}, v^1, \tilde{v})$ has weight

$$w(\tilde{C}) = \left(v^1_{f(v^1)} - v^1_a\right) + (\tilde{v}_a - \tilde{v}_{f(v^1)}) = \left(v^1_{f(v^1)} - v^1_a\right) + \underline{v} - \bar{v} \leq 0.$$

Hence, for every possible choice of $f(\tilde{v})$ we have identified a nonpositive-weight 2-cycle $\tilde{C}$ in G between 2 nodes that are allocated distinct alternatives by f. Thus we obtain a contradiction with the assumption that f is strongly monotone. □

Theorem 7 has implications in the search for local conditions which guarantee full implementability. Recall that 2-cyclicity is defined with respect to a set $\mathbf{T}$ of type profiles and a set $\mathbf{X}$ of anointment profiles. A natural direction is to restrict one or both of $\mathbf{T}$ and $\mathbf{X}$. Many choices are possible. We consider here the possibility of restricting the set of allowable anointments whilst leaving $\mathbf{T}$ unrestricted, i.e., with $\mathbf{T} = [L]^A$.

One direction is to make assumptions about (discrete) convexity properties of anointments. Consider the example in Theorem 6 where T consists of all possible type profiles. We argued that for 2-cycles to be nonnegative we would need to restrict anointments to satisfy $2x_2 - x_1 - x_3 \geq 0$. This is equivalent to requiring that anointments have decreasing differences (a notion of discrete concavity), i.e., $x_2 - x_1 > x_3 - x_2$. Recall that decreasing differences are not enough to imply the nonnegativity of all cycles, indeed $\tilde{x} = (0, 3, 5)$ has decreasing differences but yields negative 3-cycles in the anointed graph $G(\tilde{x})$.

Consider *affine anointments* for every agent i whereby $x_i(\ell) = \alpha_i \ell + \beta_i$ for some $\alpha_i, \beta_i \in \mathbb{R}$ with $\alpha_i > 0$ to guarantee that x is strictly increasing. Since conditions for full implementability involve weights of arcs which are *differences* of the $x_i(\ell)$, there is no loss in assuming that $x_i(\ell) = \ell$ for all agents i (that is, we take $\alpha = 1$ and $\beta = 0$). Under the profile x of the resulting "identity" anointment for each agent, the set of cardinal type profiles, or domain, $\mathbf{x}(\mathbf{T})$ is a consecutive integer domain. This observation leads to the following corollary of Theorem 7:

Corollary 2 *Every social choice function which is strongly* 2*-cyclic with respect to all anointment profiles* $\mathbf{X}$ *consisting entirely of affine anointments is implementable with respect to* $\mathbf{X}$.

Proof As discussed in the preceding paragraphs, it suffices to check for nonnegativity of cycles under the profile $\mathbf{x}$ of identity anointments. Since $\mathbf{x}(\mathbf{T})$ is a consecutive integer domain, by Theorem 7 we know strongly 2-cyclicity implies implementability with transfers. This implies that the anointed graph $G_{i,\mathbf{t}_{-i}}(x_i)$ has no negative weight cycles for all agents i and rival profiles $\mathbf{t}_{-i}$. Hence f is implementable with respect to $\mathbf{X}$. □

5 Discussion

In this paper we have discussed an alternate notion of "truth-telling" in the social choice setting. Our notions of full and partial implementability weaken strategy-proofness in a rational way, building on motivation from the study of mechanism design. We demonstrate that there are non-dictatorial social choice functions exhibiting full and partial implementability and, furthermore, explore some local conditions which ensure implementability.

There remain, however, several open questions. For instance, how strong a condition is full implementability? Are there natural desirable conditions that, when combined with full implementability, lead to impossibility results? Are there classes of social choices functions, other than scoring rules, which we can show are fully implementable in a wide variety of social choice settings? Further exploration is needed to answer these and related questions.

In addition, further study into how partial and full implementability relate to other measures of "degree of manipulability" (such as those of Kelly (1993) and Lepelley et al (2008)) could provide useful insights into the nature of how strategyproofness breaks down in common social choice settings.

References

Arrow KJ (1970) Social choice and individual values, vol 12. Yale University Press

Ashlagi I, Braverman M, Hassidim A, Monderer D (2010) Monotonicity and implementability. Econometrica 78(5):1749–1772

Black D (1948) On the rationale of group decision-making. The Journal of Political Economy 56(1):23–34

Boutilier C, Caragiannis I, Haber S, Lu T, Procaccia AD, Sheffet O (2012) Optimal social choice functions: A utilitarian view. In: Proceedings of the 13th ACM Conference on Electronic Commerce, pp 197–214

Caragiannis I, Procaccia AD (2011) Voting almost maximizes social welfare despite limited communication. Artificial Intelligence 175(9):1655–1671

Carroll G (2011) On mechanisms eliciting ordinal preferences, unpublished manuscript

Carroll G (2012) When are local incentive constraints sufficient? Econometrica 80(2):661–686

Gaertner W (2009) A Primer in Social Choice Theory: Revised Edition. Oxford University Press

Gibbard A (1973) Manipulation of voting schemes: a general result. Econometrica pp 587–601

Harsanyi JC (1955) Cardinal welfare, individualistic ethics, and interpersonal comparisons of utility. The Journal of Political Economy pp 309–321

Kelly JS (1993) Almost all social choice rules are highly manipulable, but a few aren't. Social choice and Welfare 10(2):161–175

Lavi R, Mu'Alem A, Nisan N (2003) Towards a characterization of truthful combinatorial auctions. In: Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on, IEEE, pp 574–583

Lepelley D, Louichi A, Smaoui H (2008) On Ehrhart polynomials and probability calculations in voting theory. Social Choice and Welfare 30(3):363–383

Mu'alem A, Schapira M (2008) Mechanism design over discrete domains. In: Proceedings of the 9th ACM Conference on Electronic Commerce, pp 31–37

Pathak PA (2011) The mechanism design approach to student assignment. Annual Review of Economics 3(1):513–536

Procaccia AD, Rosenschein JS (2006) The distortion of cardinal preferences in voting. In: Proceedings of the 10th International Workshop on Cooperative Information Agents (CIA), Springer, pp 317–331

Robbins L (1938) Interpersonal comparisons of utility: a comment. The Economic Journal 48(192):635–641

Satterthwaite MA (1975) Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. Journal of Economic Theory 10(2):187–217

Vohra R (2011) Mechanism Design: a linear programming approach. Cambridge University Press