

Finding Strategyproof Social Choice Functions via SAT Solving

Felix Brandt

*Technical University of Munich (TUM)
Munich, Germany*

BRANDTF@IN.TUM.DE

Christian Geist

*Technical University of Munich (TUM)
Munich, Germany*

GEIST@IN.TUM.DE

Abstract

A promising direction in computational social choice is to address research problems using computer-aided proving techniques. In particular with SAT solvers, this approach has been shown to be viable not only for proving classic impossibility theorems such as Arrow's Theorem but also for finding new impossibilities in the context of preference extensions. In this paper, we demonstrate that these computer-aided techniques can also be applied to improve our understanding of strategyproof irresolute social choice functions. These functions, however, require a more evolved encoding as otherwise the search space rapidly becomes much too large. Our contribution is two-fold: We present an efficient encoding for translating such problems to SAT and leverage this encoding to prove new results about strategyproofness with respect to Kelly's and Fishburn's preference extensions. For example, we show that no Pareto-optimal majoritarian social choice function satisfies Fishburn-strategyproofness. Furthermore, we explain how human-readable proofs of such results can be extracted from minimal unsatisfiable cores of the corresponding SAT formulas.

1. Introduction

Ever since the famous Four Color Problem was solved using a computer-assisted approach, it has been clear that computers can contribute significantly not only to verifying existing but also to finding and proving new results. Due to its rigorous axiomatic foundation, *social choice theory* appears to be a field in which computer-aided theorem proving is a particularly promising line of research. Perhaps the best known result in this context stems from Tang and Lin (2009), who reduce well-known impossibility results such as Arrow's theorem to finite instances, which can then be checked by a satisfiability (SAT) solver (see, e.g., Biere, Heule, van Maaren, & Walsh, 2009). Geist and Endriss (2011) were able to extend this method to a fully-automatic search algorithm for impossibility theorems in the context of preference relations over sets of alternatives. In this paper, we apply these techniques to improve our understanding of strategyproofness in the context of set-valued, or so-called irresolute, social choice functions. These types of problems, however, are more complex and require an evolved encoding as otherwise the search space rapidly becomes too large. Table 1 illustrates how quickly the number of involved objects grows and that, as a result, exhaustive search is doomed to fail.

Alternatives	4	5	6	7
Choice sets	15	31	63	127
Tournaments	64	1,024	32,768	$\sim 2 \cdot 10^6$
Canonical tournaments	4	12	56	456
Majoritarian SCFs	50,625	$\sim 10^{18}$	$\sim 10^{101}$	$\sim 10^{959}$

Table 1: Number of objects involved in problems with irresolute majoritarian SCFs

Our contribution is two-fold. On the one hand, we provide an extended framework of SAT-based computer-aided theorem proving techniques for statements in social choice theory and related research areas. Despite its complexity, this framework allows for the extraction of human-readable proofs, which eliminates the need for extensive (and difficult) verification of the underlying techniques. On the other hand, rather than only reproducing existing results, we solve some open problems, which are of independent interest, in the context of irresolute strategyproof social choice functions. These results are unlikely to have been found without the help of computers, which further strengthens the importance of the approach.

The results obtained by computer-aided theorem proving have already found attention in the social choice community (Chatterjee & Sen, 2014) and similar techniques have proven to be quite effective for other problems in economics, too. Examples are the ongoing work by Fréchette, Newman, and Leyton-Brown (2016) in which SAT solvers are used for the development and execution of the FCC’s upcoming reverse spectrum auction, recent results by Drummond, Perrault, and Bacchus (2015) who solve stable matching problems via SAT solving, as well as work by Tang and Lin (2011) who apply SAT solving to discover classes of two-player games with unique pure Nash equilibrium payoffs. In another recent paper, Caminati, Kerber, Lange, and Rowat (2015) verified combinatorial Vickrey auctions via higher-order theorem provers. In some respect, our approach bears similarities to *automated mechanism design* (see, e.g., Conitzer & Sandholm, 2002), where desirable properties are encoded and mechanisms are computed to fit specific problem instances. There is also a body of work on logical formalizations of important theorems in social choice theory, most prominently, *Arrow’s Theorem* (see, e.g., Nipkow, 2009; Grandi & Endriss, 2013; Cinà & Endriss, 2015), which has been directed more towards formalizing and verifying existing results.

Given the universality of the SAT-based method and its ease of adaptation (e.g., “testing” of similar conjectures with minimal effort by simply replacing or altering some axioms), we expect these and similar techniques to be applicable to other open problems in social choice theory and related research areas in the future. Results for different variants of the no-show paradox (Brandl, Brandt, Geist, & Hofbauer, 2015; Brandt, Geist, & Peters, 2016c) support this hypothesis. It should be noted, however, that—at least currently—an expert user or programmer is required to operate these systems. An interesting question that remains is whether it is possible to develop an automatic proof assistant that allows researchers to quickly test hypotheses on small domains without giving up too much generality and efficiency.

Let us now turn towards the social choice theoretic results. Formally, a social choice function (SCF) is defined as a function that maps individual preferences over a set of alternatives to a set of socially most-preferred alternatives. An SCF is strategyproof if no agent can obtain a more preferred outcome by misrepresenting his preferences. It is well-known from the Gibbard-Satterthwaite theorem that, when restricting attention to SCFs that always return a single alternative, only trivial SCFs can be strategyproof. The assumption of single-valuedness, however, has been criticized for being unreasonably restrictive (see, e.g., Gärdenfors, 1976; Kelly, 1977; Taylor, 2005; Barberà, 2010). A proper definition of strategyproofness for the more general setting of irresolute SCFs requires the specification of preferences over *sets* of alternatives. Rather than asking the agents to specify their preferences over all sets (which requires exponential space and would be bound to various rationality constraints), it is typically assumed that preferences over single alternatives can be extended to preferences over sets. Of course, there are various ways how to extend preferences to sets (see, e.g., Gärdenfors, 1979; Duggan & Schwartz, 2000; Taylor, 2005), each of which leads to a different class of strategyproof SCFs. A function that yields a preference relation over subsets of alternatives when given a preference relation over single alternatives is called a *set extension* or *preference extension*. In this paper, we focus on two set extensions attributed to Kelly (1977) and Fishburn (1972),¹ which have been shown to arise uniquely under very natural assumptions (Gärdenfors, 1979; Erdamar & Sanver, 2009; see also Section 2.2 of this paper).

While strategyproofness for Kelly’s extension (henceforth *Kelly-strategyproofness*) is known to be a rather restrictive condition (Kelly, 1977; Barberà, 1977; Nehring, 2000), some SCFs such as the Pareto rule, the omninomination rule, the top cycle, the uncovered set, the minimal covering set, and the bipartisan set were shown to be Kelly-strategyproof (Brandt, 2015). Interestingly, the more prominent of these SCFs are *majoritarian*, i.e., they are based on the pairwise majority relation only and can be ordered with respect to set inclusion. These results suggest that the bipartisan set may be the finest Kelly-strategyproof majoritarian SCF. In this paper, we show that this is not the case by automatically generating a Kelly-strategyproof SCF that is strictly contained in the bipartisan set. Brandt (2015) furthermore showed that, under a mild condition, Kelly-strategyproofness carries over to coarsenings of an SCF. Thus, finding inclusion-minimal Kelly-strategyproof SCFs is of particular interest. We address this problem by automating the search for these functions in small domains and report on our findings.

Existing results suggest that the more demanding notion of *Fishburn-strategyproofness* may only be satisfied by rather indiscriminating SCFs such as the top cycle (Feldman, 1979; Brandt & Brill, 2011; Sanver & Zwicker, 2012).² Using our computer-aided proving technique, we are able to confirm this suspicion by proving that, within the domain of majoritarian SCFs, Fishburn-strategyproofness is incompatible with Pareto-optimality. In order to achieve this impossibility, we manually prove a novel characterization of Pareto-optimal ma-

1. Gärdenfors (1979) attributed this extension to Fishburn because it is the weakest extension that satisfies a certain set of axioms proposed by Fishburn (1972). Some authors, however, refer to it as the Gärdenfors extension, a term which we reserve for the extension due to Gärdenfors (1976) himself.

2. The negative result by Ching and Zhou (2002) uses Fishburn’s extension but a much stronger notion of strategyproofness.

majoritarian SCFs and an induction step, which allows us to generalize the computer-generated impossibility to larger numbers of alternatives.

The paper is structured as follows. In Section 2, we present the general mathematical framework that we use throughout this paper and introduce the new condition of *tournament-strategyproofness*, which we show to be equivalent to standard strategyproofness for majoritarian SCFs. In Section 3, we describe our computer-aided proving method and explain how to encode the main questions of this paper as SAT problems. We also describe optimization techniques and other features of the approach. In Section 4, we report on our main findings—an impossibility and a possibility result—and discuss possible extensions and their limits. In Section 5, our novel approach to proof extraction from these computer-generated results is presented. We provide a human-readable proof of our main result that can be verified without the help of computers. Finally, in Section 6 we wrap up our work and give an outlook on further research directions.

2. Mathematical Framework of Strategyproofness

In this section, we provide the terminology and notation required for our results and introduce notions of strategyproofness for majoritarian SCFs that allow us to abstract away any reference to preference profiles.

2.1 Social Choice Functions

Let $N = \{1, \dots, n\}$ be a set of at least three voters with preferences over a finite set A of m alternatives. For convenience, we assume that n is odd, which entails that the pairwise majority relation is antisymmetric. The preferences of each voter $i \in N$ are represented by a complete, antisymmetric, and transitive *preference relation* $R_i \subseteq A \times A$. The interpretation of $(x, y) \in R_i$, usually denoted by $x R_i y$, is that voter i values alternative x at least as much as alternative y . The set of all preference relations over A will be denoted by $\mathcal{R}(A)$. The set of *preference profiles*, i.e., finite vectors of preference relations, is then given by $\mathcal{R}^*(A)$. The typical element of $\mathcal{R}^*(A)$ will be $R = (R_1, \dots, R_n)$. In accordance with conventional notation, we write P_i for the strict part of R_i , i.e., $x P_i y$ if $x R_i y$ but not $y R_i x$. Note that the only difference between R_i and P_i is that R_i is reflexive while P_i is not. In order to improve readability, we write $R_i : x, y, z$ as a shorthand for $x P_i y P_i z$. In a preference profile, the *weight* of an ordered pair of alternatives $w_R(x, y)$ is defined as the majority margin $|\{i \in N \mid x R_i y\}| - |\{i \in N \mid y R_i x\}|$.

Our central objects of study are *social choice functions*, i.e., functions that map the individual preferences of the voters to a nonempty set of socially preferred alternatives.

Definition 1. A *social choice function (SCF)* is a function $f : \mathcal{R}^*(A) \rightarrow 2^A \setminus \emptyset$.

An SCF is *resolute* if $|f(R)| = 1$ for all $R \in \mathcal{R}^*(A)$, otherwise it is *irresolute*.

We restrict our attention to *majoritarian SCFs*, or *tournament solutions*, which are defined using the *majority relation*. The majority relation R_M of a preference profile R is the relation on $A \times A$ defined by

$$(x, y) \in R_M \text{ if and only if } w_R(x, y) \geq 0,$$

for all alternatives $x, y \in A$. An SCF f is said to be *majoritarian* if it is neutral³ and its outcome only depends on the majority relation, i.e., $f(R) = f(R')$ whenever $R_M = R'_M$. As before, we write P_M for the strict part of R_M , i.e., $a P_M b$ if $a R_M b$ but not $b R_M a$.

An alternative x is called a *Condorcet winner* in R if $x P_M y$ for all $y \in A \setminus \{x\}$. In other words, a Condorcet winner is a “best” alternative with respect to the majority relation and it seems natural that majoritarian SCFs should select a Condorcet winner. Unfortunately, such clear-cut winners do not exist in general and a variety of so-called *Condorcet extensions*, i.e., SCFs that uniquely return a Condorcet winner whenever one exists but differ in their treatment of the remaining cases, have been proposed in the literature. In this paper, we consider the following majoritarian Condorcet extensions (see, e.g., Laslier, 1997; Brandt, Brill, & Harrenstein, 2016a, for more information).

Top Cycle Define a *dominant set* to be a non-empty set of alternatives $D \subseteq A$ such that for any alternative $x \in D$ and $y \in A \setminus D$ we have $x P_M y$. The *top cycle* TC (also known as *weak closure maximality*, *GETCHA*, or the *Smith set*) is defined as the (unique) inclusion-minimal dominant subset of A .⁴

Uncovered Set Let C denote the *covering relation* on $A \times A$, i.e., $x C y$ (“ x covers y ”) if and only if $x P_M y$ and, for all $z \in A$, $y P_M z$ implies $y P_M z$. The *uncovered set* UC contains those alternatives that are not covered according to C , i.e., $UC(R) = \{x \in A \mid y C x \text{ for no } y \in A\}$.

Bipartisan Set Consider the symmetric two-player zero-sum game in which the set of actions for both players is given by A and payoffs are defined as follows. Suppose the first player chooses a and the second player chooses b . Then the payoff for the first player is 1 if $a P_M b$, -1 if $b P_M a$, and 0 otherwise. The *bipartisan set* BP contains all alternatives that are played with positive probability in the unique Nash equilibrium of this game.

An SCF f is called a *refinement* of another SCF g if $f(R) \subseteq g(R)$ for all preference profiles $R \in \mathcal{R}^*(A)$. In short, we write $f \subseteq g$ in this case. It can be shown for the above that $BP \subseteq UC \subseteq TC$ (see, e.g., Laslier, 1997).

For our main result, we define the well-known notion of *Pareto-optimality*: an SCF f is Pareto-optimal if it never selects any *Pareto-dominated* alternative $x \in A$, i.e., $x \notin f(R)$ whenever there exists $y \in A$ such that $y P_i x$ for all $i \in N$.

2.2 Strategyproofness

Although our investigation of strategyproof SCFs is universal in the sense that it can be applied to any set extension, in this paper we will concentrate on two well-known set extensions attributed to Kelly (1977) and Fishburn (1972).⁵ These two set extensions

3. Neutrality postulates that for any permutation π of the alternatives A the SCF produces the “same” outcome (modulo the permutation). See also Section 3.1.1.

4. It is easily seen that the set of dominant sets is ordered with respect to set inclusion and therefore admits a unique minimal element. Assume for a contradiction that two dominant sets $X, Y \subseteq A$ are not contained in each other. Then, there exists $x \in X \setminus Y$ and $y \in Y \setminus X$. The definition of dominant sets requires that $x P_M y$ and $y P_M x$, a contradiction.

5. Another natural and well-known set extension by Gärdenfors leads to an even stronger notion of strategyproofness, which cannot be satisfied by any interesting majoritarian SCF (Brandt & Brill, 2011). Note

are defined as follows: Let R_i be a preference relation over A and $X, Y \subseteq A$ two nonempty subsets of A .

$$X R_i^K Y \text{ if and only if } x R_i y \text{ for all } x \in X \text{ and all } y \in Y. \quad (\text{Kelly, 1977})$$

One interpretation of this extension is that voters are completely unaware of the mechanism (e.g., a lottery) that will be used to pick the winning alternative (Gärdenfors, 1979; Erdamar & Sanver, 2009). In other words, it contains exactly the pairwise comparisons which voters can make without knowledge of the mechanism (e.g., $\{a, b\} R_i^K \{c\}$ if $a P_i b P_i c$).

$X R_i^F Y$ if and only if all of the following three conditions are satisfied:

$$\begin{aligned} &x R_i y \text{ for all } x \in X \setminus Y \text{ and } y \in X \cap Y, \\ &y R_i z \text{ for all } y \in X \cap Y \text{ and } z \in Y \setminus X, \text{ and} \\ &x R_i z \text{ for all } x \in X \setminus Y \text{ and } z \in Y \setminus X. \end{aligned} \quad (\text{Fishburn, 1972})$$

For this extension one may assume the winning alternative to be picked by a lottery according to some underlying *a priori* distribution that voters are not aware of (Ching & Zhou, 2002). Alternatively, the existence of a chairman who breaks ties according to a linear, but unknown, preference relation also rationalizes this preference extension (Erdamar & Sanver, 2009). For both of these interpretations, the extension describes exactly the conclusions a voter who is aware of the tie-breaking method can draw (e.g., $\{a, b\} R_i^F \{b, c\}$ if $a P_i b P_i c$, which does not hold for Kelly's extension R_i^K).

It is easy to see that $X R_i^K Y$ implies $X R_i^F Y$ for any pair of sets $X, Y \subseteq A$.

As we plan to prove a few results for entire classes of set extensions, we call a set extension \mathcal{E} *independent of irrelevant alternatives (IIA)* if its comparison of two sets X and Y only depends on the restriction of individual preferences to $X \cup Y$. Formally, \mathcal{E} satisfies IIA if for all pairs of preference relations R_i, R'_i and nonempty sets $X, Y \subseteq A$ such that $R_i|_{X \cup Y} = R'_i|_{X \cup Y}$ it holds that

$$X R_i^\mathcal{E} Y \text{ if and only if } X R'_i^\mathcal{E} Y.$$

This is a very mild and natural condition, which is satisfied by the previously mentioned set extensions and any other major set extension from the literature we are aware of.

Based on any set extension \mathcal{E} , we can state a corresponding notion of $P^\mathcal{E}$ -strategyproofness for irresolute SCFs. Note that in contrast to some related papers (e.g., Ching & Zhou, 2002; Sato, 2008), we interpret preference extensions as fully specified (incomplete) preference relations rather than minimal conditions on set preferences.

Again, we write $P_i^\mathcal{E}$ for the asymmetric part of $R_i^\mathcal{E}$, for any set extension \mathcal{E} .

Definition 2. Let \mathcal{E} be a set extension. An SCF f is $P^\mathcal{E}$ -manipulable by voter i if there exist preference profiles R and R' with $R_j = R'_j$ for all $j \neq i$ such that $f(R')$ is \mathcal{E} -preferred to $f(R)$ by voter i , i.e.,

$$f(R') P_i^\mathcal{E} f(R).$$

An SCF is called $P^\mathcal{E}$ -strategyproof if it is not $P^\mathcal{E}$ -manipulable.

that our negative result for Fishburn-strategyproofness trivially carries over to such more demanding set extensions.

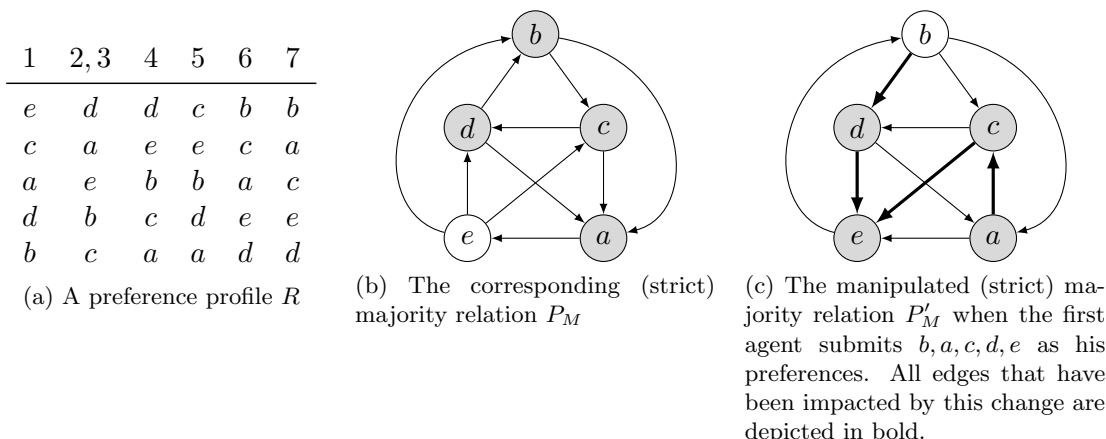


Figure 1: Let the choice sets be as indicated by shaded nodes; this example is taken from the proof of Theorem 3 (cf. Section 5.1.3). The first agent in R can P^F -manipulate by submitting b, a, c, d, e as his preferences (since $f(R') = \{a, c, d, e\} P_1^F \{a, b, c, d\} = f(R)$), but this does not constitute a P^K -manipulation (since $\{a, b, c, d\}$ and $\{a, c, d, e\}$ are incomparable according to the Kelly-extension).

It follows from the observation on set extensions above that P^F -strategyproofness implies P^K -strategyproofness. An example illustrating both notions of strategyproofness is shown in Figure 1.

Of the above SCFs, TC is P^F -strategyproof, BP is P^K - but not P^F -strategyproof, whereas UC was only known to satisfy P^K -strategyproofness (Brandt & Brill, 2011; Brandt, 2015).

2.3 Tournament-Strategyproofness

In order to allow for a more efficient encoding, we would like to omit references to preference profiles and replace them with a more succinct representation with the same expressive power. For majoritarian SCFs, the natural choice is to use the (strict) majority relation, which, for an odd number of voters, can be represented by a *tournament*:

A *tournament* is an asymmetric and complete binary relation on the set of alternatives A .⁶ We can thus view majoritarian SCFs as functions defined on tournaments rather than preference profiles, and, in slight abuse of notation,⁷ write $f(T)$ instead of $f(R)$ with $T = P_M$ being the strict part of the majority relation of R . We, furthermore, denote by $T \setminus T' := \{e \in T : e \notin T'\}$ the *edge difference* of two tournaments T and T' .

For our encoding to be efficient, it will be important to formalize the notion of strategyproofness using only references to tournaments rather than preference profiles. The

6. Note that tournaments can be defined by their edge set only. Since there is exactly one edge between any pair of vertices, the vertex set can be derived from the edge set.

7. It may be noted that, while majoritarian SCFs map from profiles (with an arbitrary, but fixed number of voters) to sets of alternatives, their interpretation via tournaments abstracts away the reference to individual voters. This has implications for Theorems 1 and 3, which depend upon the presence of a sufficient number of voters. We discuss the required number of voters in Section 5.2.

following definition serves this purpose and will be shown to be equivalent to the standard notion of strategyproofness for majoritarian SCFs.

Definition 3. A majoritarian SCF f is said to be $P^\mathcal{E}$ -*tournament-manipulable* if there exist tournaments T, T' and a preference relation $R_\mu \supseteq T \setminus T'$ such that

$$f(T') P_\mu^\mathcal{E} f(T).$$

A majoritarian SCF is called $P^\mathcal{E}$ -*tournament-strategyproof* if it is not $P^\mathcal{E}$ -tournament-manipulable.

Theorem 1. *A majoritarian SCF is $P^\mathcal{E}$ -strategyproof if and only if it is $P^\mathcal{E}$ -tournament-strategyproof.*

Proof. We show that a majoritarian SCF is $P^\mathcal{E}$ -manipulable if and only if it is $P^\mathcal{E}$ -tournament-manipulable.

For the direction from left to right, let f be a $P^\mathcal{E}$ -manipulable majoritarian SCF. Then there exist preference profiles R, R' and an integer j with $R_i = R'_i$ for all $i \neq j$ such that $f(R') P_j^\mathcal{E} f(R)$. Define tournaments $T := P_M$ and $T' := P'_M$ as the strict majority relations of R and R' , respectively. Since R and R' only differ for voter j , it follows that $T \setminus T' \subseteq R_j$, i.e., all edges that are reversed from T to T' must have been in R_j . Thus, with $R_\mu := R_j$, we get that f is tournament-manipulable.

For the converse, let f be a $P^\mathcal{E}$ -tournament-manipulable majoritarian SCF. The SCF f then admits a manipulation instance, i.e., there are two tournaments T, T' and a preference relation $R_\mu \supseteq T \setminus T'$ such that $f(T') P_\mu^\mathcal{E} f(T)$.

As in the proof of McGarvey's Theorem (McGarvey, 1953), we construct a preference profile $R^- = (R_1, \dots, R_{n-1})$ which has $T \cap T'$ as the strict part P_M^- of its majority relation: we start from an empty profile and, for each strict edge $(a, b) \in T \cap T'$, add two voters $i_{a,b}$ and $j_{a,b}$ with the preferences

$$R_{i_{a,b}} : a, b, x_1, \dots, x_{m-2} \text{ and } R_{j_{a,b}} : x_{m-2}, \dots, x_1, a, b, \text{ respectively.}$$

Here x_1, \dots, x_{m-2} denotes an arbitrary enumeration of the $m - 2$ alternatives in $A \setminus \{a, b\}$.

It then holds for the weights $w_{R^-}(a, b)$ of all edges $(a, b) \in T$ that

$$w_{R^-}(a, b) = \begin{cases} 2 & \text{if } (a, b) \in T \cap T' \\ 0 & \text{if } (a, b) \in T \setminus T'. \end{cases}$$

Note that the number of voters $n - 1$ in R^- has to be even (and at most $m^2 - m - 2$). By adding R_μ as the n -th voter, we get to a profile $R := (R^-, R_\mu)$ with an odd number of voters as required. Then $w_R(a, b) \geq 1$ for all edges $(a, b) \in T$ and, thus, R has T as its (strict) majority relation. The second profile R' can be defined to contain the same first $n - 1$ voters from R and the reversed preference $\overline{R_\mu}$ as the n -th voter (i.e., $R' := (R^-, \overline{R_\mu})$).⁸ The profile R' then has T' as its (strict) majority relation (since $w_{R'}(a, b) = -1$ for all

8. Immunity to manipulation by *reversing preferences* has been considered by Sanver and Zwicker (2012) under the name of *half-way monotonicity*. Our proof entails that (weak) half-way monotonicity is equivalent to strategyproofness for majoritarian SCFs.

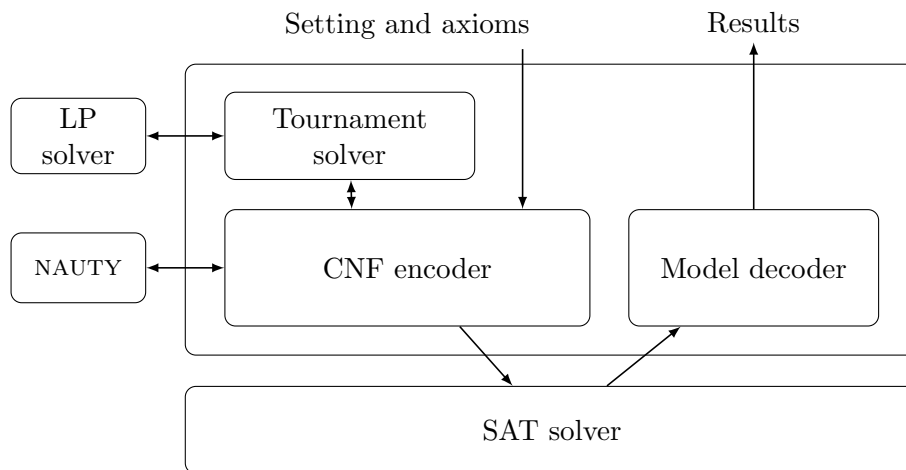


Figure 2: High-level system architecture

edges $(a, b) \in T \setminus T'$ and the weights of all edges in $T \cap T'$ are at least 1 again), which completes the manipulation instance. I.e., we have found preference profiles R, R' which only differ for voter n (who has “truthful” preferences R_μ) and for which it holds that $f(R') = f(T') \ P_\mu^\varepsilon \ f(T) = f(R)$. \square

3. Methodology

The method applied in this paper is similar to and yet more powerful than the ones presented by Tang and Lin (2009) and Geist and Endriss (2011). Rather than translating the whole problem naïvely to SAT, a more evolved approach, which resolves a large degree of freedom already during the encoding of the problem, is employed. This approach is comparable to the way SMT (satisfiability modulo theories) solving works: At the core there is a SAT solver; certain aspects of the problem, however, are dealt with in a separate theory solving unit which accepts a richer language and makes use of specific domain knowledge (Biere et al., 2009, ch. 26). The general idea, however, remains to encode the problem into a language suitable for SAT solving and to apply a SAT solver as an efficient and universal problem solving machine.

While desirable, using existing tools for higher-order formalizations directly rather than our specific approach, unfortunately, is not an option. For instance, a formalization of strategyproof majoritarian SCFs in higher-order logic (HOL) as accepted by NITPICK (Blanchette & Nipkow, 2010) is straightforward, highly flexible, and well-readable, but only successful for proofs and counterexamples involving up to three alternatives before the search space is exceeded.⁹ An optimized formalization, which we derived together with the author of NITPICK (at the cost of reduced readability and flexibility), extends the performance to four alternatives, which turns out to be just too low for our results.

9. On the other hand, the strict formalization required for NITPICK helped to identify a formally inaccurate definition of Fishburn-strategyproofness by Gärdenfors (1979) (which was later repeated by other authors).

Concretely, our approach is the following (see also the high-level architecture in Figure 2): for a given domain size n we want to check whether there exists a majoritarian SCF f that satisfies a set of axioms (e.g., P^F -strategyproofness and Pareto-optimality). We then encode the setting as well as the given axioms as a propositional formula (SAT instance) and let a SAT solver decide whether this formula has a satisfying assignment. If it has a satisfying assignment, we can decode it into a concrete instance of a majoritarian SCF f which satisfies the required properties. If the formula is unsatisfiable, we know that no such function f exists.

As we will see, depending on the problem, some preparatory tasks have to be solved before the actual encoding: (i) sets, tournaments, and preference relations are enumerated; (ii) isomorphisms between tournaments are determined using the tool NAUTY (McKay & Piperno, 2013); (iii) choice sets for specific SCFs are computed (e.g., via matrix multiplication for UC and linear programming for BP).

In the following, we describe in more detail how the general setting of majoritarian SCFs as well as desirable properties, such as strategyproofness, can be encoded as a SAT problem in CNF (conjunctive normal form).¹⁰ First, we describe our initial encoding, which is expressive enough to encode all required properties, but allows for small domain sizes of (depending on the axioms) at most four to five alternatives only. Second, we explain how this encoding can be optimized to increase the overall performance by orders of magnitude such that larger instances of up to seven alternatives are solvable.

3.1 Initial Encoding

By design, SAT solvers operate on propositional logic. A direct and naïve propositional encoding of the problem would, however, require a huge number of propositional variables since many higher-order concepts are involved (e.g., sets of alternatives, preference relations over sets as well as over alternatives, and functions from tuples of such relations to sets). In our approach, we use only one type of variable to encode SCFs. The variables are of the form $c_{T,X}$ with T being a tournament and X being a set of alternatives.¹¹ The semantics of these variables are that $c_{T,X}$ if and only if $f(T) = X$, i.e., the majoritarian SCF f selects the set of alternatives X as the choice set for any preference profile with (strict) majority relation T . In total, this gives us a high but manageable number of $2^{\frac{m(m-1)}{2}} \cdot 2^m = 2^{\frac{m(m+1)}{2}}$ variables in the initial encoding.

An encoding with variables $c_{T,x}$ for alternatives x rather than sets would require less variable symbols. This encoding, however, leads to much more complexity in the generated clauses, which more than offsets these savings. This imbalance is best exhibited in the encoding of strategyproofness where statements are always made for pairs of outcomes (i.e., sets of alternatives). Each occurrence of $c_{T,X}$ could be replaced by $\bigwedge_{x \in X} c_{T,x} \wedge \bigwedge_{y \notin X} \neg c_{T,y}$. But since this formula then contains a conjunction within a disjunction, which is not possible

10. Converting an arbitrary propositional formula naïvely to CNF can lead to an exponential blow-up in the length of the formula. There are, however, well-known efficient techniques (e.g., Tseitin’s encoding, see Tseitin, 1983) to avoid this at the cost of introducing linearly many auxiliary variables. We apply these techniques manually when needed.

11. In all algorithms, a subroutine $c(T, X)$ will take care of the compact enumeration of variables. Since we know in advance how many tournaments and non-empty subsets there are, we can simply use a standard enumeration method for pairs of objects.

in CNF, either expansion (and therefore an exponential blow-up) or replacement (e.g., by a helper variable $c_{T,X} \leftrightarrow \bigwedge_{x \in X} c_{T,x}$) would be required.

The following two subsections demonstrate the initial encoding of both contextual as well as explicit axioms to CNF.

3.1.1 CONTEXT AXIOMS

Apart from the explicit axioms, which we are going to describe in the next subsection, there are further axioms that need to be considered in order to fully model the context of majoritarian SCFs. For this purpose, an arbitrary function that maps tournaments to non-empty sets of its vertices will be called a *tournament choice function*. Using our initial encoding three axioms are introduced, which will ensure that functionality of the tournament choice function and neutrality are respected (making it a tournament solution): (1) functionality, (2) canonical isomorphism equality, and (3) the orbit condition.

The first axiom ensures that the relational encoding of f by variables $c_{T,X}$ indeed models a function rather than an arbitrary relation, i.e., for each tournament T there is exactly one set X such that the variable $c_{T,X}$ is set to true. In formal terms this can be written as

$$\begin{aligned} & (\forall T) ((\exists X) c_{T,X} \wedge (\forall Y, Z) Y \neq Z \rightarrow \neg(c_{T,Y} \wedge c_{T,Z})) \\ \equiv & \bigwedge_T \left(\left(\bigvee_X c_{T,X} \right) \wedge \bigwedge_{Y \neq Z} (\neg c_{T,Y} \vee \neg c_{T,Z}) \right). \end{aligned} \quad (1)$$

As an illustrative example, the corresponding simple pseudo-code for generating the CNF file can be found in Appendix B.

The second and third axiom together constitute neutrality of the tournament choice function f , which, formally, can be written as

$$\pi(f(T)) = f(\pi(T)) \text{ for all tournaments } T \text{ and permutations } \pi : A \rightarrow A.$$

A direct encoding of this neutrality axiom, however, would be tedious due to the quantification over all permutations. In addition, our reformulation as *canonical isomorphism equality* and *orbit condition* enables a substantial optimization of the encoding as we will see in Section 3.2. We require further observations in order to precisely state these two axioms.

We use the well-known fact that graph isomorphisms define an equivalence relation on the set of all tournaments.¹² For each equivalence class, pick a representative as the *canonical tournament* of this class. For any tournament T , we then have a unique canonical representation (denoted by T_c). We also pick one of the potentially many isomorphisms from T_c to T as the *canonical isomorphism* of T and denote it by π_T .¹³ This allows us to formulate the axiom of *canonical isomorphism equality*.

Definition 4. A tournament choice function f satisfies *canonical isomorphism equality* if

$$f(T) = \pi_T(f(T_c)) \text{ for all tournaments } T. \quad (2)$$

12. Two tournaments T and T' are isomorphic if there is a permutation $\pi : A \rightarrow A$ such that $\pi(T) = T'$.

13. In practice, the tool NAUTY will automatically compute canonical representations for both tournaments and isomorphisms.

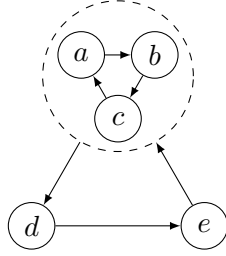


Figure 3: The orbits of this tournament are $\mathcal{O}_T = \{\{a, b, c\}, \{d\}, \{e\}\}$. A corresponding automorphism would be $\alpha = \begin{pmatrix} a & b & c & d & e \\ b & c & a & d & e \end{pmatrix}$. $C := \{a, b, c\}$ represents a component in the sense that for all of its elements $x \in C$ it holds that $x P_M d$ and $e P_M x$.

For the last of the three context axioms, the definition of an *orbit* should be clarified. The orbits of a tournament T are equivalence classes of alternatives according to the following equivalence relation: two alternatives a, b are considered equivalent if and only if there is an *automorphism* $\alpha : A \rightarrow A$ which maps a to b , i.e., for which $\alpha(a) = b$. The set of orbits of a tournament T is denoted by \mathcal{O}_T . An example can be found in Figure 3.

Definition 5. A tournament choice function f satisfies the *orbit condition* if

$$O \subseteq f(T_c) \text{ or } O \cap f(T_c) = \emptyset \quad (3)$$

for all canonical tournaments T_c and their orbits $O \in \mathcal{O}_{T_c}$.

It can be shown that for any tournament choice function, neutrality is equivalent to the conjunction of the orbit condition and canonical isomorphism equality, or equivalently, that the class of tournament choice functions satisfying the orbit condition and canonical isomorphism equality is equal to the class of tournament solutions. We formalize this statement in Lemma 1. The proof of Lemma 1 is based on standard arguments from category theory and is presented in Appendix A.

Lemma 1. *For any tournament choice function, neutrality is equivalent to the conjunction of the orbit condition and canonical isomorphism equality.*

3.1.2 EXPLICIT AXIOMS

Many axioms can be efficiently encoded in our proposed encoding language. In this section we present the main conditions required to achieve the results in Section 4. Clearly, the most important one is strategyproofness. In formal terms, P^ε -tournament-strategyproofness can be written as

$$\begin{aligned} & (\forall T, T', R_\mu \supseteq T \setminus T') \neg \left(f(T') P_\mu^\varepsilon f(T) \right) \\ & \equiv \bigwedge_T \bigwedge_{T'} \bigwedge_{R_\mu \supseteq T \setminus T'} \bigwedge_{Y P_\mu^\varepsilon X} (\neg c_{T,X} \vee \neg c_{T,Y}) \end{aligned} \quad (4)$$

where T, T' are tournaments, R_μ is a preference relation, and X, Y are non-empty subsets of A . The algorithmic encoding of strategyproofness is omitted here since we present an optimized version in Section 3.2.

Another property of SCFs that will play an important role in our results is the one of being a refinement of another (known) SCF g . Fortunately, this can easily be encoded using our framework:

$$\begin{aligned} & (\forall T)(\exists X \subseteq g(T)) f(T) = X \\ \equiv & \bigwedge_T \bigvee_{X \subseteq g(T)} c_{T,X}. \end{aligned} \quad (5)$$

If we desire that the resulting SCF f is different from g (for instance, to obtain a *strict* refinement in conjunction with Axiom (5)), we encode the additional clause:

$$\begin{aligned} & (\exists T) f(T) \neq g(T) \\ \equiv & \bigvee_T \neg c_{T,g(T)}. \end{aligned} \quad (6)$$

Finally, even properties regarding the cardinalities of choice sets can be encoded. The following axiom—stating that $|f(T)| < |g(T)|$ for at least one tournament T —will, for instance, be useful in Section 4.1.1 when searching for SCFs that return small choice sets:

$$\begin{aligned} & (\exists T)(\exists X) |X| < |g(T)| \wedge f(T) = X \\ \equiv & \bigvee_T \bigvee_{\substack{X \\ |X| < |g(T)|}} c_{T,X}. \end{aligned} \quad (7)$$

3.2 Optimized Encoding for Improved Performance

In order to efficiently solve instances of more than four alternatives, we need to streamline our initial encoding without weakening its logical and expressiv power. In this section, we present the three optimization techniques we found most effective.

3.2.1 OBVIOUS REDUNDANCY ELIMINATION

A straightforward first step is to reduce the obvious redundancy within the axioms. As an example, consider the axiom of strategyproofness, where—in order to determine whether an outcome $Y = f(T')$ is preferred to an outcome $X = f(T)$ —we consider *all* preference relations $R_\mu \supseteq T \setminus T'$. It suffices, however, if we stop after finding the first such preference relation with $Y P_\mu^\varepsilon X$ because then we already know that not both $Y = f(T')$ and $X = f(T)$ can be true.

Similarly, in many axioms, we can exclude considering symmetric pairs of objects (e.g., for functionality of the tournament choice function, there is no need to consider both pairs of sets (X, Y) and (Y, X)).

3.2.2 CANONICAL TOURNAMENTS

The main efficiency gain can be achieved by making use of the canonical isomorphism equality (see Section 3.1.1) during encoding. Recall that this condition states that for any tournament T the choice set $f(T)$ can be determined from the choice set $f(T_c)$ of the corresponding *canonical tournament* T_c by applying the respective canonical isomorphism π_T .

```

foreach Canonical tournament  $T_c$  do
  foreach Tournament  $T'$  do
     $R_{T_c \setminus T'} \leftarrow \{R_\mu \mid R_\mu \text{ is a preference relation and } R_\mu \supseteq T_c \setminus T'\};$ 
    foreach Set  $X$  do
      foreach Set  $Y$  do
        boolean found  $\leftarrow$  false;
        foreach  $R_\mu \in R_{T_c \setminus T'}$  do
          if  $\neg \textit{found} \wedge \textit{setExt}(R_\mu, \mathcal{E}).\textit{prefers}(Y, X)$  then
            variable\_not( $c(T_c, X)$ );
            variable\_not( $c(T'_c, \pi_{T'}^{-1}(Y))$ );
            newClause();
            found  $\leftarrow$  true;

```

Algorithm 1: $P^\mathcal{E}$ -tournament-strategyproofness (optimized)

Therefore, it suffices to formulate the axioms on a single representative of each equivalence class of tournaments, in our case, the canonical tournament. The magnitudes in Table 1 illustrate that this formulation dramatically reduces the required number of variables, the size of the CNF formula, and the time required for encoding it.

In particular, in all axioms we can replace any outer quantifier $\forall T$ by a quantifier $\forall T_c$ that ranges over canonical tournaments only.¹⁴ In the case of strategyproofness, however, there is a second tournament T' for which the restriction to canonical tournaments is potentially not strong enough to capture the full power of the axiom. We therefore keep T' as an arbitrary tournament but make sure that we only need variable symbols $c_{T'_c, Y}$ for canonical tournaments in our CNF encoding. This can be achieved through the canonical isomorphism $\pi_{T'}$ since by Condition (2), $f(T') = Y$ if and only if $f(T'_c) = \pi_{T'}^{-1}(Y)$. The optimized encoding is shown in Algorithm 1.

Furthermore, since we no longer make any statements within the CNF formula about non-canonical tournaments, the canonical isomorphism equality condition becomes an “empty” condition and, thus, can be dropped from the encoding.

3.2.3 APPROXIMATION THROUGH LOGICALLY RELATED PROPERTIES

Approximation is a standard tool in SAT/SMT which can speed up the solving process. For instance, over-approximation can help find unsatisfiable instances faster by only solving parts of the full problem description in CNF. If this partial CNF formula is found to be unsatisfiable, any superset will also trivially be unsatisfiable. Since common manipulation instances in the literature require only one edge in a tournament to be reversed, one can, for instance, use over-approximation in the form of *single-edge-strategyproofness*, a slightly *weaker* variant of (tournament-)strategyproofness with $|T \setminus T'| = 1$.¹⁵

14. The tool NAUTY is capable of enumerating such non-isomorphic (i.e., canonical) tournaments.

15. While it was not obvious whether this condition is actually strictly weaker than tournament-strategyproofness, we identified Pareto-optimal SCFs that are Kelly-single-edge-strategyproof but not Kelly-tournament-strategyproof (cf. Section 4.1.1).

If the solver returns that there is no single-edge-strategyproof SCF that satisfies some set of properties Γ , we know immediately that there is also no strategyproof SCF that satisfies Γ . We used this form of approximation to prove the results in Remark 2.¹⁶

In a similar fashion, one can also apply *logically simpler* conditions, such as the ones by Brandt and Brill (2011), that are slightly stronger or weaker than $P^{\mathcal{E}}$ -strategyproofness for specific set extensions \mathcal{E} in order to *logically* under- or over-approximate problems, respectively. While these logically simpler conditions can help to further improve encoding and solving times, none of them were required to obtain the results presented in this paper.

Another way to over-approximate our problems is to restrict the domain of the SCF (e.g., by random sampling), which we explore in somewhat more detail when extracting small proofs in Section 5.1.1.

3.3 Finding Refinements through Incremental Solving

In order to obtain results for most refined (i.e., inclusion-minimal) or otherwise minimal SCFs, it will be important to also produce this property to the SAT solver in a satisfactory way. Generally, since the task of a SAT solver is to generate only one satisfying assignment, it does not necessarily output the finest SCF to satisfy a given set of properties. Through iterated or incremental solving, however, we can force the SAT solver to generate progressively finer or simply different SCFs that satisfy a set of desired properties.¹⁷ For refinements, this can be achieved by adding clauses which encode that the desired SCF must be (strictly) finer than previously found solution (see, e.g., the formulation in Section 3.1.2). When the finest SCF with the desired properties has been found, adding these clauses leads to an unsatisfiable formula, which the SAT solver detects and therefore verifies the minimality of the solution.

With this final solving step, we have the main tools at hand required for our results, the most significant ones of which we describe in the next section.

4. Results and Discussion

Here we present our two main findings:

- There exists a strict refinement of BP which is $P^{\mathbf{K}}$ -strategyproof (Theorem 2).
- For majoritarian SCFs with $m \geq 5$, $P^{\mathbf{F}}$ -strategyproofness and Pareto-optimality are incompatible (Theorem 3). For $m < 5$, UC satisfies $P^{\mathbf{F}}$ -strategyproofness and Pareto-optimality.

Further minor results are mentioned in the discussions preceding the proofs and in Section 4.2.1.

16. While for $m = 7$ approximation was required to reach the result, it also enabled a speed-up for smaller instances: the running time for $m = 6$, for example, was reduced from almost five hours to three minutes.

17. Note that finding a refinement of an SCF is *not* equivalent to finding a smaller/minimal model in the SAT sense; in our encoding all assignments have the same number of satisfied variables.

4.1 Minimal Kelly-Strategyproof SCFs

Brandt (2015) showed that every coarsening f of a P^K -strategyproof SCF f' is P^K -strategyproof if $f(R) = f'(R)$ whenever $|f'(R)| = 1$. Thus, it is an interesting question to identify finest (or *inclusion-minimal*) P^K -strategyproof SCFs.

While previous results suggested that BP could be a—or even *the*—finest majoritarian SCF which satisfies P^K -strategyproofness, we first provide a counterexample to these assertions using $m = 5$ alternatives, and second show that also for larger domain sizes there exist majoritarian refinements of BP that are still P^K -strategyproof and return significantly smaller choice sets than BP .

Theorem 2. *There exists a majoritarian Condorcet extension that refines BP and is still P^K -strategyproof. As a consequence, BP is not even a finest majoritarian Condorcet extension satisfying P^K -strategyproofness.*

Proof. Within seconds our implementation finds a satisfying assignment for $m = 5$ and the encoding of the explicit axioms *refinement of BP* (implies *Condorcet extension*) and *P^K -strategyproofness*. The corresponding majoritarian SCF can be decoded from the assignment and is defined like BP with the exception depicted in Figure 4. \square

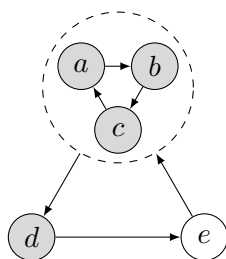


Figure 4: Tournament on which a P^K -strategyproof refinement of BP is possible. $C := \{a, b, c\}$ represents a component in the sense that for all of its elements $x \in C$ it holds that $x P_M d$ and $e P_M x$. While BP chooses the whole set A on this tournament, the refined solution selects $\{a, b, c, d\}$ only.

Using the technique described in Section 3.3, we furthermore confirmed that the obtained SCF is the only refinement of BP on five alternatives which is still P^K -strategyproof. Note, however, that it does not satisfy the (natural, but strong) property of *composition-consistency* (see, e.g., Laslier, 1997). Thus, it remains open whether BP might be characterized as an—or even *the*—inclusion-minimal, P^K -strategyproof, and composition-consistent majoritarian SCF.¹⁸

While we were not able to resolve this open problem completely, we proved the following statements by extending our approach to also cover composition-consistency. BP is an inclusion-minimal, P^K -strategyproof, and composition-consistent majoritarian SCF

18. Although already on the domain of up to five alternatives there are further inclusion-minimal, P^K -strategyproof, and composition-consistent Condorcet extensions, which we could find using the computer-aided method, these counterexamples might not extend to larger domains.

for $m \leq 5$.¹⁹ For $m \leq 7$, BP is an inclusion-minimal majoritarian SCF satisfying *set-monotonicity*²⁰ and composition-consistency. While this result might extend to larger instances, it only holds for at most 5 alternatives that these properties *uniquely* characterize BP .

If we, however, drop composition-consistency again, we can find multiple inclusion-minimal majoritarian SCFs that are refinements of BP and still P^K -strategyproof. Interestingly, some of these SCFs turn out to be more discriminating than others in the sense that on average they yield significantly smaller choice sets. In the following section we are going to search for such discriminating SCFs and analyze the average size of their respective choice sets.

4.1.1 FINDING DISCRIMINATING KELLY-STRATEGYPROOF SCFs

Many P^K -strategyproof tournament solutions have been criticized for not being discriminating enough. It is known, for instance, that in large random tournaments, TC and UC select all alternatives with probability approaching 1 (Scott & Fey, 2012), while BP selects exactly half of the alternatives on average for any fixed number of alternatives (Fisher & Reeves, 1995). More discriminating tournament solutions, on the other hand, such as the *Copeland*, *Markov*, and *Slater* rules violate P^K -strategyproofness. Using the computer-aided approach, we search for the most discriminating majoritarian SCFs that satisfy P^K -strategyproofness. Though this is in the spirit of automated mechanism design (see, e.g., Conitzer & Sandholm, 2002), we apply these techniques mostly to improve our understanding of P^K -strategyproofness and related axioms rather than to propose the generated tournament solutions for actual use.

As a measure for the discriminating power of majoritarian SCFs, we use the *average relative size* $\text{avg}(f)$ of the choice sets returned by an SCF f . Formally we define

$$\text{avg}(f) := \frac{1}{|A| \cdot |\mathcal{T}|} \sum_{T \in \mathcal{T}} |f(T)|,$$

where \mathcal{T} is the set of all labeled tournaments on $|A| = m$ alternatives. We call an SCF f more discriminating than another SCF g if $\text{avg}(f) < \text{avg}(g)$. Given a set of axioms Γ , we try to find a *most discriminating* SCF f (i.e., with the minimal value for $\text{avg}(f)$) such that f satisfies the axioms in Γ .

While in theory it would be possible to just encode the relevant axioms and then enumerate all SCFs with the required properties by incrementally applying Axiom (6), the number of such SCFs is usually much too large. If we instead refine the initial solution further and further by applying Axioms (5) and (6) as indicated in Section 3.3, we will find an inclusion-minimal SCF, but not necessarily a most discriminating SCF f . We thus proceed via Algorithm 2, which is guaranteed to find a most discriminating SCF f without enumerating *all* candidates of SCFs. The algorithm starts by constructing an initial candidate of an SCF which satisfies the required axioms, iteratively refines it as much as possible (via the conjunction of Axioms (5) and (6)), and then encodes an additional axiom stating

19. For $m = 6$ we can already find a refinement with the same properties.

20. Set-monotonicity postulates that the choice set is invariant under the weakening of unchosen alternatives; it implies P^K -strategyproofness (Brandt, 2015).

that all future solutions must yield a choice set with strictly smaller cardinality for at least one tournament T (Axiom (7)). The algorithm then repeats the refinement and encoding process until no further solution can be found. Since Axiom (7) is a necessary condition for $\text{avg}(f) < \text{avg}(g)$, we can be sure that a finest SCF f is returned.

```

SCF smallestSolution  $\leftarrow$  null;
CNF minimalRequirements  $\leftarrow$  encodeAxioms();
minimalRequirements  $\leftarrow$  preprocess(minimalRequirements); // optional
while isSatisfiable(minimalRequirements) do
  CNF currentRequirements  $\leftarrow$  minimalRequirements;
  SCF currentSolution  $\leftarrow$  solve(currentRequirements);
  while canBeRefined(currentSolution) do
    Append Axioms (5) and (6) to currentRequirements with  $g = \text{currentSolution}$ ;
    currentSolution  $\leftarrow$  solve(currentRequirements);
  // an inclusion-minimal solution has been found
  if avgSize(currentSolution)  $<$  avgSize(smallestSolution) then
    smallestSolution  $\leftarrow$  currentSolution;
  Append Axiom (7) to minimalRequirements with  $g = \text{currentSolution}$ ;
return smallestSolution;

```

Algorithm 2: A search algorithm to find a cardinality-minimal SCF f (i.e., with minimal value for $\text{avg}(f)$) that satisfies a given set of axioms. As a reminder, Axioms (5) and (6) encode a strict refinement of g ; Axiom (7) encodes $|f(T)| < |g(T)|$ for some tournament T .

Preprocessing is generally optional in Algorithm 2; for $m = 6$ we, however, had to use unit propagation in order to reduce the size of the resulting SAT instance.²¹ Note that the optimization techniques as described in Section 3.2 (in particular, canonical tournaments) can also be applied here.

The results of our analysis are exhibited in Figure 5. While on up to four alternatives all axioms under consideration lead to the same minimal size of $\text{avg}(f)$, on larger domains, $P^{\mathbf{K}}$ -strategyproofness allows for smaller choice sets than BP (e.g., 45% instead of 50% of the alternatives for $m = 6$). Interestingly, the gap between BP and these more discriminating SCFs that satisfy $P^{\mathbf{K}}$ -strategyproofness is not extraordinarily large; in particular, moving from $P^{\mathbf{K}}$ -strategyproofness to $P^{\mathbf{K}}$ -single-edge-strategyproofness allows for a more sizable reduction of $\text{avg}(f)$. For the related property of Kelly-participation, Brandl et al. (2015) remarked that the average size of choice sets can be reduced by almost 50% compared to BP , which supports the intuition that participation is a “weaker” property than strategyproofness (even though logically the two are independent).

BP and set-monotonicity yield the exact same values of $\text{avg}(f)$ for $m \leq 6$, which is somewhat surprising as we found SCFs that are not coarsenings of BP and are yet set-monotonic on this domain size. These SCFs, however, have no set-monotonic refinements that are more discriminating than BP . Interestingly, this does not generalize to larger

21. For the case of Kelly-strategyproofness, unit propagation and deletion of duplicate clauses reduced the CNF formula from about 600 million to just below three million clauses.

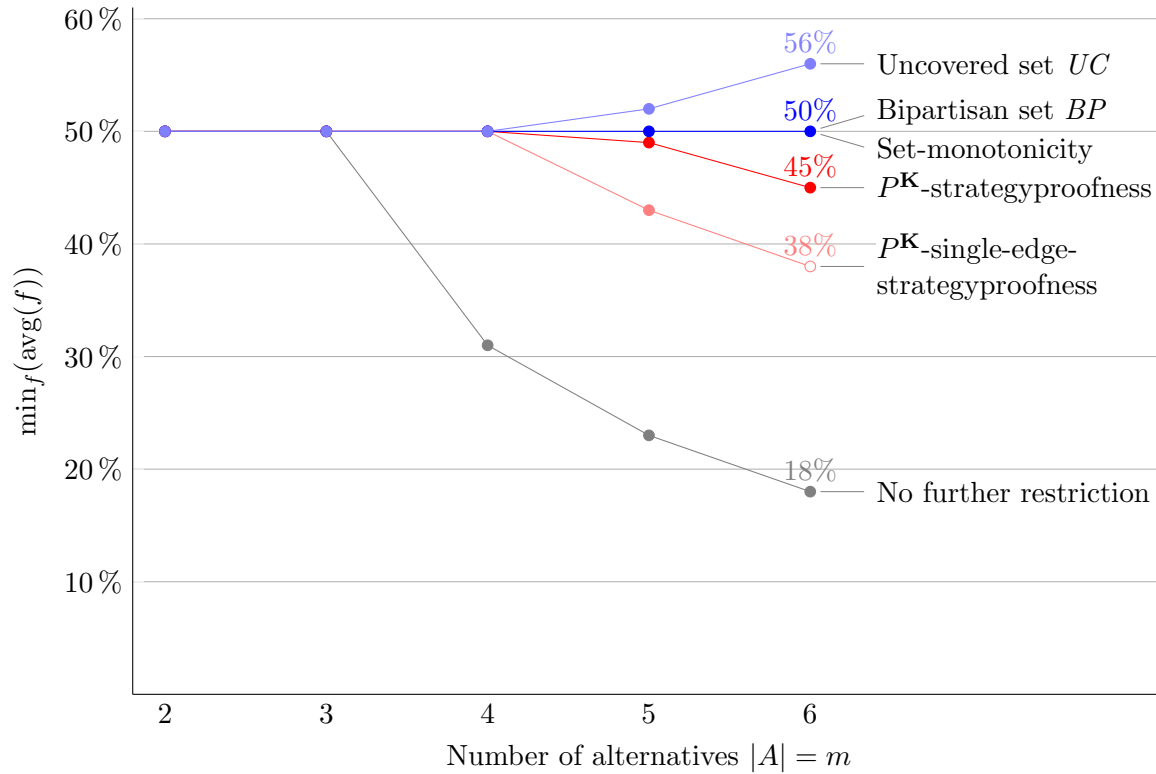


Figure 5: A comparison of the minimal values (rounded) of $\text{avg}(f)$ for majoritarian, Pareto-optimal SCFs f that satisfy the given axioms (e.g., P^K -strategyproofness). Interestingly, the values for *set-monotonicity* are identical to the ones for *BP*. Non-solid dots represent upper bounds, i.e., cases where we could only compute an SCF f with this value of $\text{avg}(f)$ but have no guarantee that it is indeed minimal.

domains since we found a most discriminating majoritarian SCF f for $m = 7$ that satisfies set-monotonicity and Pareto optimality while only selecting 49.73% of the alternatives on average.

As more demanding axioms usually lead to larger choice sets (for instance, the SCF that always returns all alternatives trivially satisfies many axioms), one might view the minimal value of $\text{avg}(f)$ as an attempt to “quantify” the strength of an axiom. We leave a more detailed study of such a quantification as future work.

4.2 Incompatibility of Fishburn-Strategyproofness and Pareto-Optimality

In order to prove our main result on the incompatibility of Pareto-optimality and $P^{\mathbf{F}}$ -strategyproofness we first show the following lemma, which establishes that, for majoritarian SCFs, the notion of Pareto-optimality is equivalent to being a refinement of the uncovered set (UC).²²

Lemma 2. *A majoritarian SCF f is Pareto-optimal if and only if it is a refinement of UC .*

Proof. It is well-known, and was already observed by Fishburn (1977), that UC is Pareto-optimal, which implies that all its refinements are also Pareto-optimal.

For the direction from left to right, let f be a Pareto-optimal majoritarian SCF and T an arbitrary tournament. It suffices to show that $f(T)$ can never contain a covered alternative (since then $f(T) \subseteq UC(T)$ contains uncovered alternatives only). So let b be an alternative that is covered by another alternative a . We are going to construct a preference profile R which has T as its (strict) majority relation and in which b is Pareto-dominated by a . Together with the Pareto-optimality of f this implies that $b \notin f(T)$. We use a variant of the well-known construction by McGarvey (1953), but for triples rather than pairs of alternatives. Note that for each voter we need to ensure that he strictly prefers a to b in order to obtain the desired Pareto-dominance of a over b . Starting with an empty profile, for each alternative $x \notin \{a, b\}$ we add two voters R_{x_1}, R_{x_2} to the profile. These two voters are defined depending on how x is ranked relative to a and b in order to establish the edges between a, x and b, x . Note that since $x T a$ implies $x T b$ (because of $a C b$), edge (a, b) cannot be contained in a three-cycle with x and, thus, forms a transitive triple with x .

- Case 1: $x T a$ (implies $x T b$)
 $R_{x_1} : x, a, b, v_1, \dots, v_{m-3}; \quad R_{x_2} : v_{m-3}, \dots, v_1, x, a, b$
- Case 2a: $a T x$ and $x T b$
 $R_{x_1} : a, x, b, v_1, \dots, v_{m-3}; \quad R_{x_2} : v_{m-3}, \dots, v_1, a, x, b$
- Case 2b: $a T x$ and $b T x$
 $R_{x_1} : a, b, x, v_1, \dots, v_{m-3}; \quad R_{x_2} : v_{m-3}, \dots, v_1, a, b, x$

Here v_1, \dots, v_{m-3} denotes an arbitrary enumeration of the $m - 3$ alternatives in $A \setminus \{a, b, x\}$.

In all cases, the two voters cancel out each other for all pairwise comparisons other than (a, b) , (x, a) and (x, b) . For each of the remaining edges $(y, z) \in T$ (with $\{y, z\} \cap \{a, b\} = \emptyset$)

²². A stronger version of this lemma was shown by Brandt, Geist, and Harrenstein (2016b).

we further add two voters (now even closer to the construction by McGarvey.)

$$R_{(y,z)_1} : y, z, a, b, v_1, \dots, v_{m-4} \text{ and}$$

$$R_{(y,z)_2} : v_{m-4}, \dots, v_1, a, b, y, z,$$

which together establish edge (y, z) , reinforce (a, b) and cancel otherwise. Note that in order to achieve an odd number of voters, an arbitrary voter can be added without changing the majority relation (as all edges had a weight of at least two so far). This completes the construction of a preference profile R which has T as its (strict) majority relation and in which b is Pareto-dominated by a . \square

To establish the full result (which does not admit a proof by counterexample as in Theorem 2) we—similarly to previous approaches—make use of an inductive argument.

Lemma 3. *For any set extension \mathcal{E} that satisfies IIA, if there exists a majoritarian SCF f for $m + 1$ alternatives that is $P^\mathcal{E}$ -strategyproof and Pareto-optimal, then there also exists a majoritarian SCF f' for just m alternatives that satisfies these two properties.*

Proof. Let $f \subseteq UC$ be a majoritarian SCF for $m + 1 \geq 2$ alternatives that is $P^\mathcal{E}$ -strategyproof. Then we define f_e to be the restriction of f to m alternatives based on tournaments in which alternative e is a Condorcet loser, i.e., an alternative x for which $(y, x) \in T$ for all $y \in A \setminus \{x\}$. In formal terms, define

$$f_e(T) := f(T^{+e}),$$

where T^{+e} is the tournament obtained from T by adding an alternative e as a Condorcet loser. This restriction of f is a well-defined SCF since alternative e cannot be contained in $f(T^{+e}) \subseteq UC(T^{+e}) = UC(T)$, where the last equation follows from the simple observation that the covering relation is unaffected by deleting Condorcet losers.

We now need to show that for some alternative e the restriction f_e is a majoritarian SCF that is $P^\mathcal{E}$ -strategyproof and Pareto-optimal. Since this holds for any $e \in A$, we just pick e arbitrarily.

- **Majoritarian:** The fact that f_e is a majoritarian SCF carries over trivially from f .
- **$P^\mathcal{E}$ -strategyproofness:** Assume for a contradiction that f_e is not $P^\mathcal{E}$ -strategyproof. Then, by Theorem 1 there exist tournaments T and T' on m alternatives such that $f_e(T') P_\mu^\mathcal{E} f_e(T)$ with $R_\mu \supseteq T \setminus T'$. But since $f_e(T') = f(T'^{+e})$ and $f_e(T) = f(T^{+e})$ (and by the fact that \mathcal{E} satisfies IIA), we get

$$f(T'^{+e}) P_\mu^\mathcal{E} f(T^{+e}),$$

which contradicts $P^\mathcal{E}$ -tournament-strategyproofness of f (as the two tournaments T'^{+e} and T^{+e} form a manipulation instance), and thus $P^\mathcal{E}$ -strategyproofness.

- **Pareto-optimality:** By Lemma 2, this is equivalent to being a refinement of UC . Thus, let T be an arbitrary tournament on m alternatives and consider the following chain of set inclusions, which proves that $f_e \subseteq UC$:

$$f_e(T) = f(T^{+e}) \subseteq UC(T^{+e}) = UC(T).$$

□

By virtue of Lemma 3 it now suffices to check the claim for the restricted domain of $m = 5$, which we do in the following lemma.

Lemma 4. *For exactly five alternatives (i.e., $m = 5$) there is no majoritarian SCF f that satisfies $P^{\mathbf{F}}$ -strategyproofness and Pareto-optimality.*

Proof. This base case of $m = 5$ alternatives was verified using our computer-aided approach, i.e., we checked that, with $|A| = 5$ alternatives, there is no satisfying assignment for an encoding of $P^{\mathbf{F}}$ -tournament-strategyproofness (cf. Theorem 1) and being a refinement of UC (cf. Lemma 2), which the SAT solver confirmed within seconds. A human-readable proof of this claim has been extracted from the computer-aided approach and is presented in Section 5.1.2. □

Finally, this paper’s main result regarding $P^{\mathbf{F}}$ -strategyproofness follows directly from Lemmas 3 and 4.

Theorem 3. *For any number of alternatives $m \geq 5$ there is no majoritarian SCF f that satisfies $P^{\mathbf{F}}$ -strategyproofness and Pareto-optimality.*

Proof. We prove the statement inductively. The base case of $m = 5$ is covered by Lemma 4. For the induction step, we apply the contrapositive of Lemma 3 with $\mathcal{E} := \mathbf{F}$, which directly yields the desired results. □

While the number of voters required for this impossibility has been kept implicit so far, an upper bound of at most $m^2 - m - 1 = 19$ voters can be derived from the construction in the proof of Theorem 1. In Section 5 we will see, however, that a human-readable proof of Theorem 3 can be extracted, which only requires seven voters.

As a consequence of Theorem 3, virtually all common tournament solutions—except the top cycle (see Remark 2)—fail to be $P^{\mathbf{F}}$ -strategyproof.

4.2.1 REMARKS

Before we turn towards the technique of proof extraction, let us discuss some further insights regarding Theorem 3, which have been, to a large extent, enabled by the universality of the presented method.

Remark 1 (Strengthenings). It can be shown with the computer-aided method that Theorem 3 holds even without the assumption of neutrality. Since then, however, the optimizations based on canonical tournaments can no longer be used, extracted proofs (cf. Section 5) are much more complex and we therefore decided to present the result with neutrality here.²³

The theorem can be further strengthened by additionally only requiring $P^{\mathbf{F}}$ -single-edge-strategyproofness (cf. Section 3.2) or an even weaker variant of $P^{\mathbf{F}}$ -strategyproofness where the manipulator is only allowed to swap two adjacent alternatives (see, e.g., Sato, 2013).

23. In addition, running times are much longer, which, however, is not a major concern given that not many conjectures had to be tested for this result.

Remark 2 (The Top Cycle TC). Note that Theorem 3 is not in conflict with the fact that TC is P^F -strategyproof, as, for $m \geq 4$ alternatives, TC is strictly coarser than UC and therefore not Pareto-optimal. Possibly, TC is even the finest majoritarian Condorcet extension that satisfies P^F -strategyproofness for $m \geq 5$. We were able to verify this for $5 \leq m \leq 7$ using our computer program. In the case of four alternatives, UC is a strict refinement of TC and (as our method shows) still P^F -strategyproof. For $m = 8$ the time and space requirements appear to be prohibitive; already for $m = 7$ (despite all optimizations and approximations) encoding and solving the problem takes almost 24 hours, while for $m = 6$ it runs in about three minutes. It is not obvious whether an inductive argument can extend these verified instances to larger numbers of alternatives (as, for instance, such an induction step would require at least five alternatives).

Remark 3 (Other Preference Extensions). An advantage of the computer-aided approach is its universality. We can, for instance, very easily adapt the implementation to check set extensions other than the ones by Kelly and Fishburn.

Interestingly, our main result only relies on a small fraction of the power of the Fishburn extension: it suffices to only compare disjoint sets and sets that are contained in one another. In formal terms, the following set extension suffices for the impossibility:

$$X R_i^{F^-} Y \text{ if and only if } \begin{cases} X R_i^K Y & \text{when } X \cap Y = \emptyset, \\ X R_i^F Y & \text{when } X \subseteq Y \text{ or } Y \subseteq X, \\ \perp & \text{otherwise.} \end{cases}$$

Actually, it would even suffice to only compare sets X and Y such that $|X \cap Y| \leq 3$.

We also checked a strengthening of the Fishburn extension: a voter prefers a set X to a set Y if X is better than Y under both optimistic and pessimistic expectations.

Formally, $X R_i^{OP} Y$ if and only if

$$\begin{aligned} &x R_i y \text{ for all } x \in X \text{ and some } y \in Y, \text{ and} \\ &y R_i x \text{ for all } y \in Y \text{ and some } x \in X. \end{aligned}$$

This extension is a weakening of both the optimistic and the pessimistic notions of strategyproofness in the Duggan-Schwartz Theorem (Duggan & Schwartz, 2000). In the majoritarian setting, P^{OP} -strategyproofness leads to an analogous impossibility as in Theorem 3 for $m \geq 4$ already.

Remark 4 (Generality of Lemma 3). Note that the proofs of the individual properties within the inductive proof of Lemma 3 do only rely on the definition of f_e and stand independently of each other. Furthermore, it may be noted that Lemma 3 can even be shown for refinements of arbitrary majoritarian SCFs g whose choice set $g(T)$ does not shrink when Condorcet losers are removed from T (rather than Pareto-optimal majoritarian SCFs).

5. Proof Extraction

A major concern regarding computer-aided proofs is the difficulty of checking their correctness. While our implementation correctly confirmed a number of existing results and this

can be considered as testing, some doubts about the correctness of new results naturally remain. Most SAT solvers offer some kind of proof trace, which can be checked by third-party-software. This, however, does not guarantee correctness of the encoding but only confirms the unsatisfiability of the corresponding CNF formula.

In this section, we show how *human-readable* proofs of the desired statements can be extracted from our approach, which can then be verified just as any manual mathematical proof. The general idea of this proof extraction technique lies in finding and analyzing a *minimal unsatisfiable core* (also referred to as a *minimal unsatisfiable set* (MUS)) of the SAT instance. An unsatisfiable core of a CNF formula is a subset of clauses that is already unsatisfiable by itself. If any subset of clauses of the unsatisfiable core is satisfiable, then the core is called minimal. In our case, the minimal unsatisfiable core contains information about the concrete instances of axioms that have to be employed to obtain an impossibility (e.g., manipulation instances, applications of Pareto optimality, etc). This information can be extracted in a straightforward way and reveals the structure and arguments of the proof.

We exemplify this technique in Section 5.1, in which we extract a human-readable proof of our main result (Theorem 3). In Section 5.2 we additionally enrich this proof by a set of *minimal* corresponding preference profiles, which then shows that the result of Theorem 3 holds for any setting with at least seven voters.

In general, extracting human-readable proofs serves two separate purposes. On the one hand, a human-readable proof can significantly raise confidence in the correctness of the results, basically by making verification of the *approach* obsolete since now the *results* themselves are directly verifiable. On the other hand, the extracted proofs sometimes provide additional insight into the problems via their arguments and structure. In our case, the number of voters required for the impossibility would not have been (easily) accessible directly.

5.1 A Human-Readable Proof of Theorem 3

In order to extract a human-readable proof of Theorem 3, or actually its main ingredient Lemma 4, we have to follow a series of three steps:

1. Obtain a suitable MUS of the CNF formula that encodes a $P^{\mathbf{F}}$ -tournament-strategyproof refinement of UC on five alternatives
2. Decode the MUS into a human-readable format
3. Interpret the human-readable MUS to obtain a human-readable proof

While the first two steps are computer-aided and can be largely automated, step three requires some manual effort.

5.1.1 OBTAINING A SUITABLE MUS OF THE CNF FORMULA

Extracting a minimal unsatisfiable core is a feature offered by a range of SAT solvers. In this paper, we use PICOMUS (part of PICOSAT, Biere, 2008) for this job.²⁴ It should be

24. Compiled with trace support in order to use core extraction in addition to clause selector variables. This significantly improves the size of the resulting MUS.

noted, however, that while an MUS is inclusion-minimal, it does not necessarily represent a *smallest* unsatisfiable set (i.e., with a minimal number of clauses or variables).²⁵

As the number of clauses turned out to be a good proxy for proof complexity and length, we tried to find an MUS with a small number of clauses. When run on the complete, optimized SAT encoding as described in Section 3.2, PICOMUS returns an MUS with 55 clauses. This is already a massive reduction compared to more than three million clauses in the original problem instance, but we found an even smaller MUS with only *16 clauses* by randomly sampling sets of tournaments to be used instead of the full domain of all tournaments when generating our problem files. Another heuristic approach of considering “neighborhoods” of single tournaments (for instance, all tournaments that can be reached by changing at most two edges in the transitive tournament) yielded a less significant improvement with a total of 25 clauses.

While it seems natural that larger domains are generally better as they lead to the required impossibility more often than smaller domains, larger domains actually tend towards larger proofs and even miss very small proofs. For instance, for the domain size $s = 200$ (consisting of s labeled tournaments) no proof smaller than 18 clauses was found, while the same number of runs with $s = 50$ produced four proofs with just 16 clauses each.²⁶

Therefore, in our setting, a medium-sized domain ($s = 50$ or $s = 100$ in our experiments) appears to be best suited. The complete results of running time and proof size analysis given different domain sizes s can be obtained from Figures 8 and 9 in Appendix C.

5.1.2 DECODING THE MUS INTO A HUMAN-READABLE FORMAT

The next step is to make the obtained MUS more accessible to humans. To this end, we first (automatically) add comments to the original CNF for each manipulation clause during its creation, and then select those comments that belong to clauses in the MUS. The comments contain witnesses for the manipulation instances found, i.e., information about the original tournament T , the manipulated tournament T' , the respective choice sets $f(T)$ and $f(T')$, and the original preferences of the manipulator R_μ (compare Definition 3). Furthermore, any variable symbol can easily be decoded into the tournament and choice set it represents, which is helpful in particular for all non-manipulation clauses (orbit condition and Pareto-optimality).

The result of this step is presented in Figure 6, where each tournament is represented by a lower triangular representation of its adjacency matrix (see the proof of Lemma 4 in Section 5.1.3 for graphical representations).

5.1.3 INTERPRETING THE MUS AND OBTAINING A HUMAN-READABLE PROOF

From the witnessed MUS it is just a small step to a textual, human-readable proof. With a bit of practice, one can quickly understand the structure of the proof: it starts from the orbit condition in the first line and the refinement condition in the last line, which each

25. While the tool CAMUS by Liffiton and Sakallah (2008) is theoretically capable of finding a *smallest* MUS (with a minimal number of clauses), it did not terminate in a reasonable amount of time on our very large CNF instances.

26. In addition, medium-sized domains are more efficient regarding their running time per generated proof, which admittedly plays only a minor, but still important role given that the total running time for large domains is about 20 hours.

```

p cnf 341 16
218 231 232 233 234 247 248 0
-202 -330 0
c T: 1111111111 -> [e]; T': 1011100111 -> [d,e]; P_i: b,d,c,e,a
-233 -202 0
c T: 1101100111 -> [e]; T': 0010100111 -> [a]; P_i: b,c,d,a,e
-234 -202 0
c T: 1101100111 -> [a,e]; T': 0010100111 -> [a]; P_i: b,c,d,a,e
-218 -218 0
c T: 1101100111 -> [a]; T': 1001000100 -> [e]; P_i: e,c,a,d,b
-232 -232 0
c T: 1101100111 -> [a,b,c,d]; T': 1001000100 -> [a,c,d,e]; P_i: e,c,a,d,b
-248 -338 0
c T: 1101100111 -> [a,b,c,d,e]; T': 1100100101 -> [b,c,e]; P_i: b,e,c,d,a
-231 -202 0
c T: 1101100111 -> [b,c,d]; T': 1111111111 -> [e]; P_i: a,e,b,c,d
-247 -202 0
c T: 1101100111 -> [b,c,d,e]; T': 1111111111 -> [e]; P_i: a,e,b,c,d
-314 -314 0
c T: 1100101110 -> [c]; T': 1100100101 -> [e]; P_i: b,d,e,a,c
c T: 1100101110 -> [c]; T': 1100110110 -> [b]; P_i: b,c,d,e,a
-318 -318 0
c T: 1100101110 -> [d]; T': 1100100101 -> [b]; P_i: b,d,e,a,c
c T: 1100101110 -> [d]; T': 1100110110 -> [a]; P_i: b,c,e,a,d
-322 -322 0
c T: 1100101110 -> [c,d]; T': 1100110110 -> [a,b]; P_i: b,e,a,c,d
-326 -326 0
c T: 1100101110 -> [e]; T': 1100110110 -> [d]; P_i: b,c,d,e,a
-334 -202 0
c T: 1100101110 -> [d,e]; T': 1001111010 -> [d]; P_i: c,a,d,e,b
202 0
314 318 322 326 330 334 338 0

```

Figure 6: A version of the extracted MUS, in which all manipulation instances (here: binary clauses) have been decoded into a human-readable format: two mappings of tournaments (original T and manipulated T') to choice sets and the truthful preferences of the manipulator P_μ . This information covers all variables and thus suffices to also decode the remaining clauses.

Truthful choice	Manipulated choice	Manipulator's preferences	
$f(T_1) = \begin{cases} \{e\} & \not\subseteq \\ \{a\} \cup \{e\} & \not\subseteq \\ \{b, c, d\} & \not\subseteq \\ \{b, c, d\} \cup \{e\} & \not\subseteq \\ \{a\} & \not\subseteq \\ \{a\} \cup \{b, c, d\} & \not\subseteq \end{cases}$	$f(T_a) \subseteq UC(T_a) = \{a\}$	b, c, d, a, e	
	$f(T_e) \subseteq UC(T_e) = \{e\}$	a, e, b, c, d	
	$f(T'_1) = \begin{cases} \{e\} & T'_1 \text{ is iso-} \\ \{a, c, d, e\} & \text{morphic}^{27} \text{ to } T_1 \end{cases}$		e, c, a, d, b
$f(T_2) = \begin{cases} \{c\} & \not\subseteq \\ \{d\} & \not\subseteq \\ \{c, d\} & \not\subseteq \\ \{e\} & \not\subseteq \\ \{d, e\} & \not\subseteq \end{cases}$	$f(T'_2) = \begin{cases} \{b\} & T'_2 \text{ is iso-} \\ \{a\} & \text{morphic}^{27} \text{ to } T_2 \\ \{a, b\} & \\ \{d\} & \\ f(T_d) \subseteq UC(T_d) = \{d\} \end{cases}$	b, e, a, c, d	
		$f(T_d) \subseteq UC(T_d) = \{d\}$	b, c, d, e, a
		$f(T_e) \subseteq UC(T'_e) = \{e\}$	c, a, d, e, b
		$f(T_2) = \{c, e\} \not\subseteq^{28}$	a, c, b, e, d

Table 2: Set of manipulation instances (one per line) to conclude that $f(T_1) = A = \{a, b, c, d, e\}$ and $f(T_2) = \{c, d, e\}$. Each of the truthful choices considered here leads to a $P^{\mathbf{F}}$ -tournament-manipulation instance (a contradiction to the assumption of $P^{\mathbf{F}}$ -tournament-strategyproofness). The tournaments are defined in Figure 7.

leave some (limited) possibilities for respective choice sets, and then excludes all possible choices one after another by suitable manipulation instances. The full proof runs as follows.

Proof of Lemma 4. For a contradiction, let f be a majoritarian SCF on $A = \{a, b, c, d, e\}$ that satisfies $P^{\mathbf{F}}$ -strategyproofness and Pareto-optimality. Recall that, by Theorem 1, f is $P^{\mathbf{F}}$ -tournament-strategyproof, too, and by Lemma 2 it has to be a refinement of UC (i.e., $f \subseteq UC$). Let furthermore T_1 and T_2 be the tournaments depicted in Figure 7. We proceed in three steps: first, we show that $f(T_1) = UC(T_1) = A$. Second, we argue that $f(T_2) = UC(T_2) = \{c, d, e\}$. And last, we prove that these two insights actually forms the basis of a manipulation instance, which leads to the desired contradiction.

Let us start with $f(T_1) = UC(T_1) = A$. First, note that since the alternatives $\{b, c, d\}$ form an orbit we know that either $\{b, c, d\} \subseteq f(T_1)$ or $\{b, c, d\} \cap f(T_1) = \emptyset$ (cf. Definition 5). We are going to exclude all remaining choice sets through $P^{\mathbf{F}}$ -tournament-manipulation instances. As a first example, suppose $f(T_1) = \{e\}$. Then a voter with individual preferences $P_\mu : b, c, d, a, e$ could reverse the edges (b, a) and (b, c) in T_1 such that a transitive tournament T_a with Condorcet winner a results (which needs to be uniquely selected by f since $f \subseteq UC$). Since, however, $\{a\} P_\mu^{\mathbf{F}} \{e\}$, this contradicts $P^{\mathbf{F}}$ -tournament-strategyproofness. The same example also works to exclude $f(T_1) = \{a, e\}$. Note how these arguments correspond to lines 5 to 8 of the extracted MUS in Figure 6. The (analogous) manipulation instances for all possible choice sets other than $A = \{a, b, c, d, e\}$ are given in Table 2 and Figure 7.

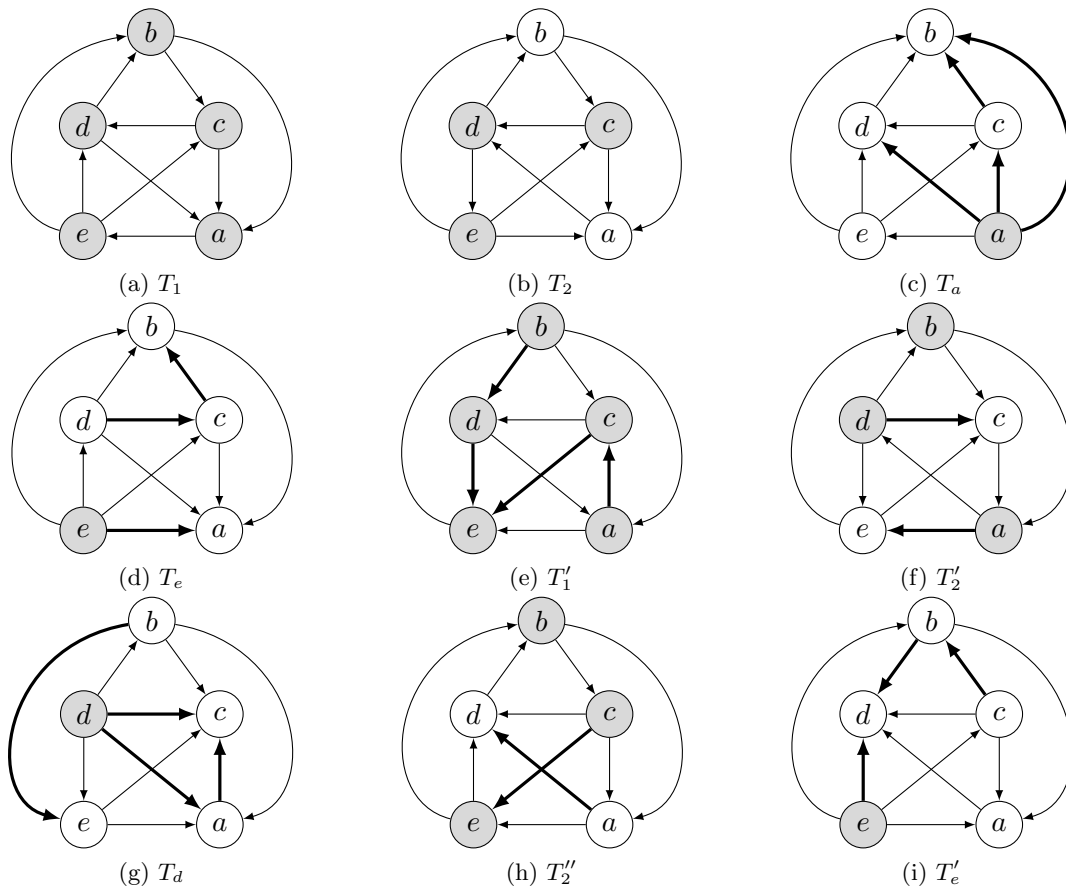


Figure 7: Tournaments which are required in the proof of Lemma 4. The uncovered sets are marked in grey; edges that have been (for T'_e : will be) reversed by the manipulating voter (cf. Table 2) are depicted as thick edges. Note the proof would also succeed with less edge reversals in T_a , T_e , T_d , and T'_e (such that these tournaments only have Condorcet winners rather than being transitive). These transitive tournaments are isomorphic, however, and thus can be succinctly represented as the single clause 202 in the extracted MUS.

For $f(T_2) = UC(T_2) = \{c, d, e\}$, first observe that $f(T_2) \subseteq UC(T_2) = \{c, d, e\}$ and hence we only need to exclude any strict subset of $\{c, d, e\}$. Again we proceed by giving a possible manipulation instance for each of those subsets. The complete list is to be found in Table 2 and Figure 7. Observe how the last line in Table 2 excludes $f(T_2) = \{c, e\}$ by considering it as the *manipulated* choice for the (known) truthful choice $f(T'_e) \subseteq UC(T'_e) = \{e\}$.

As a last step, we provide a manipulation instance based on $f(T_1) = A$ and $f(T_2) = \{c, d, e\}$. For this, first observe that by renaming the alternatives we get $f(T''_2) = \{b, c, e\}$ and so the manipulation instance results from a voter with preferences $P'_\mu : b, e, c, d, a$. This

27. The isomorphisms are $\pi_1 = \begin{pmatrix} a & b & c & d & e \\ b & e & c & d & a \end{pmatrix}$ and $\pi_2 = \begin{pmatrix} a & b & c & d & e \\ d & c & a & e & b \end{pmatrix}$, respectively.

28. The SAT solver actually returned an isomorphic copy of this instance, which we restructured to improve readability.

voter can reverse the edges (d, a) and (e, c) in T_1 to create T_2'' and obtain the $P^{\mathbf{F}}$ -preferred outcome $\{b, c, e\}$, a contradiction to the $P^{\mathbf{F}}$ -strategyproofness of f . \square

Note that actually only the manipulation instance with $f(T_1) = \{a\} \cup \{b, c, d\}$ and $f(T_1') = \{a, c, d, e\}$ requires the Fishburn-extension; for the other instances the Kelly-extension suffices.

5.2 Number of Voters Required

In the previous parts of the paper we have taken advantage of the fact that our condition of *tournament-strategyproofness* abstracted away any reference to voters. It is interesting to ask, however, how many voters are at least required for the obtained impossibility of Theorem 3 to hold. The construction in the proof of Theorem 1 gives an implicit upper bound of $m^2 - m - 1 = 19$ voters, but this can be further improved to *seven* voters.

By slightly modifying the techniques described by Brandt, Geist, and Seedig (2014), we were able to (automatically) construct *minimal* preference profiles for all steps in Proof 5.1.3. While Brandt et al. (2014) provided a SAT-formulation of whether a given majority relation can be induced by a given number of voters, we extended this framework to include axioms for manipulation instances. In more detail, we re-used the axioms for *linear preferences* and majority implications, but added axioms for the *truthful preferences of the manipulator* and *majority implications for the manipulated profile*.

The profiles that we generated for all steps in the proof of Lemma 4 in Section 5.1.3 are given in Appendix D. The largest of these profiles contains seven voters, and all other profiles can easily be extended to seven voters by adding pairs of voters with opposite preferences. While this observation shows that seven is the smallest number of voters which can be achieved with our extracted proof, it remains open whether, by another proof, the number of voters can be further reduced below seven.

6. Conclusion

We have extended and applied computer-aided theorem proving based on SAT solving to extensively analyze Kelly- and Fishburn-strategyproof majoritarian SCFs. This has led to a range of results, both positive and negative. An important novel contribution of our work is the ability to extract a human-readable proof from negative SAT instances. This eliminates the need to verify the computer-aided method since impossibility results can directly be checked based on their human-readable proofs. Based on the ease of adaptation of the proposed method, we anticipate further insights to spring from the overall approach in the future. Apart from simply applying our system to further investigate strategyproofness, other potential applications related to our line of work include:

Unrestricted SCFs In order to reduce complexity, we have studied *majoritarian* SCFs only. The framework, however, is applicable in the same way to general SCFs, which “operate” on full preference profiles (rather than majority relations). The challenge then is to find a suitable representation of such preference profiles and potentially corresponding inductive arguments on the number of voters.

Further axioms Some preliminary experiments suggest that our technique can easily be applied to a range of properties other than strategyproofness, these deserve further investigation. In many cases it suffices to just formalize and implement the additional axioms. Of particular interest could be such properties that link the behavior of SCFs for different domain sizes. As initial steps in this direction, we were able to extend the approach to cover the property of *participation* (Brandl et al., 2015; Brandt et al., 2016c) as well as a weak version of *composition-consistency* (cf. Section 4.1).

Smallest number of voters required As mentioned in Section 5.2, Theorem 3 holds for any number of voters $n \geq 7$, but it is not known whether this number is minimal. One could adapt proof extraction as presented in Section 5 to search for a smallest proof in the number of voters, rather than in the number of clauses, to settle this question.

Generalization of the inductive argument It appears reasonable to investigate whether the inductive argument of Lemma 3 can be further generalized to a whole class of properties/axioms, ideally based on their logical form. As in the work of Geist and Endriss (2011), this would then enable an automated search for further theorems about SCFs.

Apart from these concrete ideas, applications of the general approach can be envisioned in many areas of theoretical economics.

Acknowledgments

This material is based upon work supported by Deutsche Forschungsgemeinschaft under grants BR 2312/7-2 and BR 2312/9-1. The paper benefitted from discussions at the COST Action Meeting IC1205 on Computational Social Choice (Maastricht, 2014), the 13th International Conference on Autonomous Agents and Multiagent Systems (Paris, 2014), the 5th International Workshop on Computational Social Choice (Pittsburgh, 2014), and the Dagstuhl Seminar on Computational Social Choice: Theory and Applications (Dagstuhl, 2015). The authors in particular thank Jasmin Christian Blanchette, Markus Brill, Hans Georg Seedig, and Bill Zwicker for helpful discussions and their support, and three anonymous reviewers for their valuable comments and suggestions to improve the paper.

Appendix A. Proof of Lemma 1

We first show that the orbit condition is equivalent to a statement about automorphisms:

Lemma 5. *Let f be a tournament choice function. Then the following statement is equivalent to the orbit condition:*

$$\alpha(f(T_c)) = f(T_c) \text{ for all canonical tournaments } T_c \text{ and their automorphisms } \alpha. \quad (8)$$

Proof. Let f be a tournament choice function and T_c a canonical tournament. For the direction from left to right, let furthermore $O \in \mathcal{O}_{T_c}$ an orbit on T_c . Now pick two alternatives $a, b \in O$. We show that either both alternatives are chosen by f or neither one is. Since a and b are in the same orbit, there must be an automorphism α on T_c for which $\alpha(a) = b$. Observe that $a \in f(T_c)$ if and only if $b \in \alpha(f(T_c))$ if and only if $b \in f(T_c)$, where the last step is an application of Condition (8).

For the converse, let α be an automorphism on T_c , pick an arbitrary alternative $a \in A$ and consider its inverse image $\alpha^{-1}(a) =: b$. Since a and b are in the same orbit, it holds by the orbit condition that $a \in f(T_c)$ if and only if $b \in f(T_c)$. Furthermore, as $\alpha(b) = a$ we get that $a \in f(T_c)$ if and only if $a \in \alpha(f(T_c))$. Thus, $f(T_c) = \alpha(f(T_c))$, which is what we wanted to prove. \square

Next we prove a general statement about how to split any isomorphism into a canonical isomorphism and an automorphism.

Lemma 6. *Any isomorphism $\pi : T_c \rightarrow T$ can be decomposed into the canonical isomorphism π_T and an automorphism $\alpha : T_c \rightarrow T_c$. I.e., for any isomorphism $\pi : T_c \rightarrow T$ there is an automorphism $\alpha : T_c \rightarrow T_c$ such that $\pi = \pi_T \circ \alpha$.*

Proof. Define $\alpha : T_c \rightarrow T_c$ by setting $\alpha := \pi_T^{-1} \circ \pi$. Since inverses and compositions of isomorphisms are themselves isomorphisms, it follows directly that α is an automorphism. Furthermore, $\pi_T \circ \alpha = \pi_T \circ (\pi_T^{-1} \circ \pi) = (\pi_T \circ \pi_T^{-1}) \circ \pi = \pi$. \square

Lemmas 5 and 6 together can then be used to prove Lemma 1:

Lemma 1. *For any tournament choice function, neutrality is equivalent to the conjunction of the orbit condition and canonical isomorphism equality.*

Proof. Let f be a tournament choice function and first note that by Lemma 5 we might use Condition (8) rather than the orbit condition. Therefore, the direction from left to right is trivially true.

For the direction from right to left, we first only show that canonical isomorphism equality (2) together with Condition (8) implies neutrality *for canonical tournaments*: So let T_c be a canonical tournament, π a permutation and define $T' := \pi(T_c)$. By Lemma 6, we can decompose the isomorphism $\pi : T_c \rightarrow T'$ such that $\pi = \pi'_T \circ \alpha$ for some automorphism α on T_c . Then the following chain of equalities holds, which proves the claim for canonical tournaments:

$$f(\pi(T_c)) = f(T') \stackrel{(2)}{=} \pi'_{T'}(f(T'_c)) = \pi'_{T'}(f(T_c)) \stackrel{(8)}{=} \pi'_{T'}(\alpha(f(T_c))) = \pi(f(T_c)). \quad (9)$$

For arbitrary tournaments T and permutations π , we write T as $\pi_T(T_c)$ and obtain

$$f(\pi(T)) = f(\pi(\pi_T(T_c))) = f((\pi \circ \pi_T)(T_c)),$$

which, since T_c is canonical, is equal to

$$(\pi \circ \pi_T)(f(T_c)) = \pi(\pi_T(f(T_c))) \stackrel{(2)}{=} \pi(f(T))$$

by Condition (9). This finishes the proof. \square

Appendix B. Pseudo-Code for Encoding

We present (as an illustrative example) the simple pseudo-code of Algorithm 3 to generate the CNF form of Axiom 1 (functionality of the tournament choice function; cf. Section 3.1.1).

```

foreach Tournament  $T$  do
  foreach Set  $X$  do
     $\lfloor$  variable(c(T, X));
  newClause();
  foreach Set  $Y$  do
    foreach Set  $Z \neq Y$  do
       $\lfloor$  variable_not(c(T, Y));
       $\lfloor$  variable_not(c(T, Z));
       $\lfloor$  newClause();

```

Algorithm 3: Functionality of the tournament choice function

Appendix C. MUS Search Analysis (Running Time and Size of MUS)

In this appendix, we present the complete results of the running time (Figure 8) and MUS size (measured in number of clauses; Figure 9) analyses given different sizes s of randomly sampled domains. In our setting, sizes of $s = 50$ or $s = 100$ appear to offer good results both in terms of running time and actually finding small proofs.

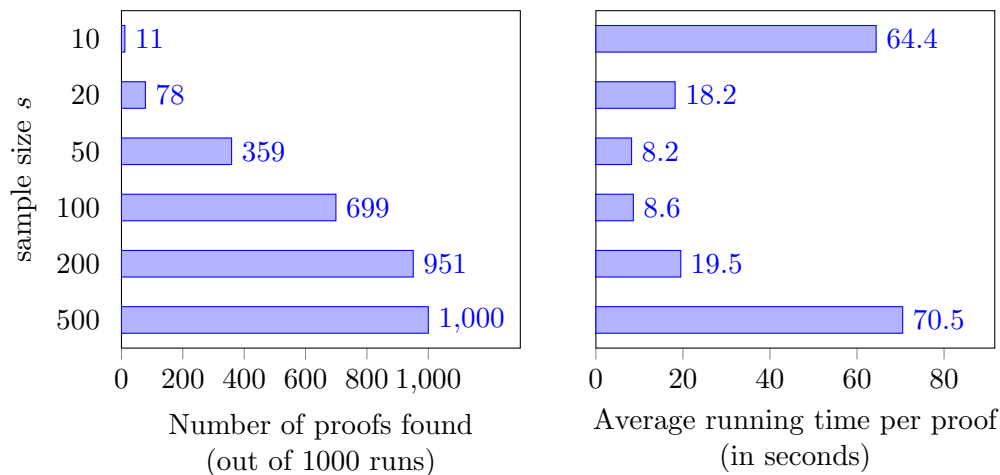


Figure 8: Number of unsatisfiable instances (i.e., proofs found) and running time results under heuristics with different numbers s of sampled tournaments (labeled, 1000 runs).

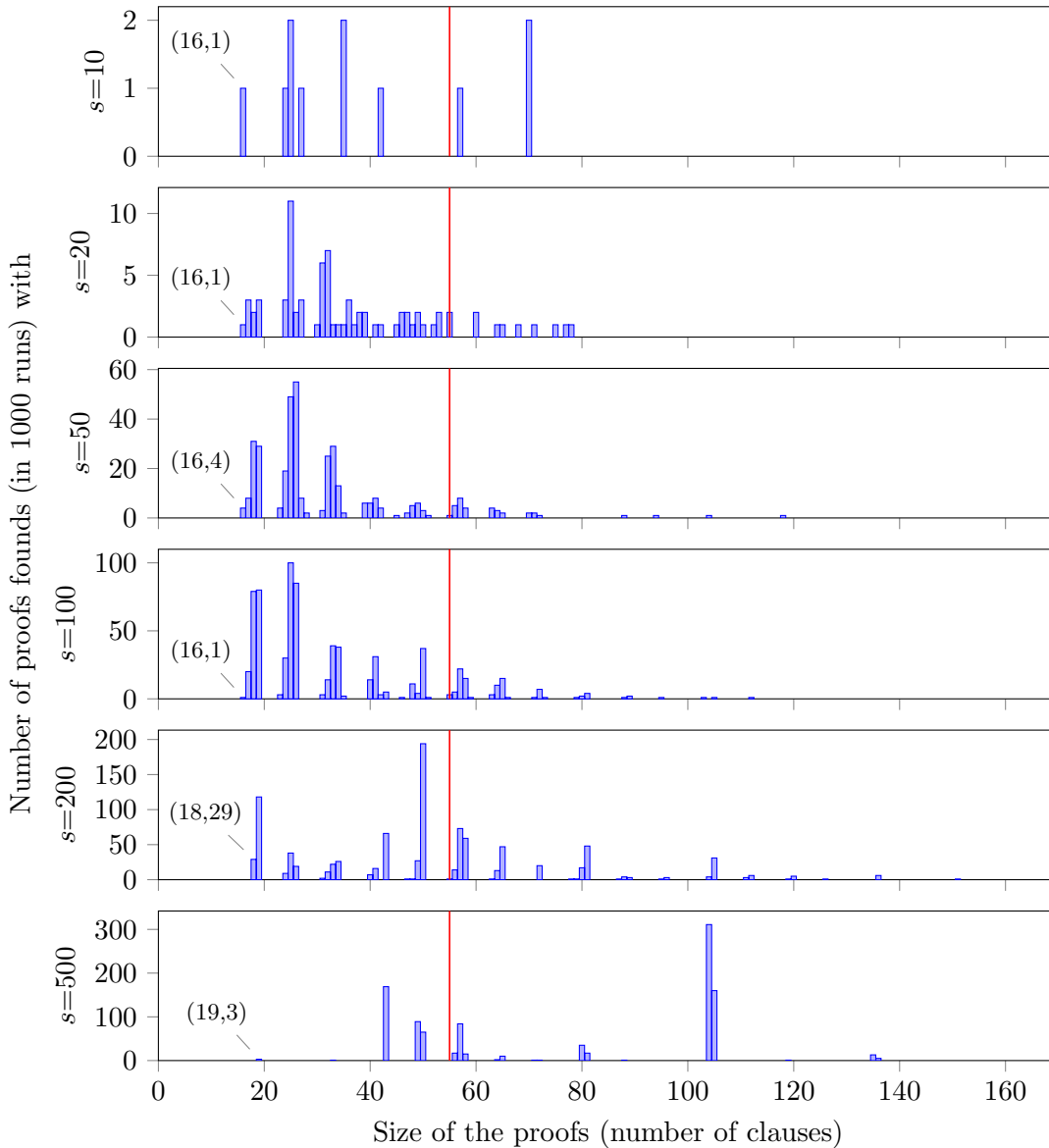


Figure 9: The sizes of MUSes (proofs) under heuristics with different numbers s of sampled tournaments (labeled). The size of the MUS obtained from running on the full domain is indicated by a red line. For improved readability, the size and multiplicity of the smallest MUS is explicitly listed.

Appendix D. Profiles for the Extracted Proof of Theorem 3

Here we display the MUS of Figure 6 enriched with *minimal* preference profiles for each step in the proof of Theorem 3. The profiles were generated and checked for minimality on a computer (and using a SAT solver) in less than a second each.

```

p cnf 341 16
218 231 232 233 234 247 248 0
Agent 0: b,c,d,a,e
Agent 1: a,e,c,d,b
Agent 2: e,d,b,c,a

-202 -330 0
c T: 1111111111 -> [e]; T': 1011100111 -> [d,e]; P_i: b,d,c,e,a
Agent 0: b,d,c,e,a
Agent 1: c,d,a,e,b
Agent 2: e,c,d,b,a
Agent 3: a,e,d,c,b
Agent 4: e,d,b,a,c
Manipulated preferences of agent 0: b,a,c,d,e

-233 -202 0
c T: 1101100111 -> [e]; T': 0010100111 -> [a]; P_i: b,c,d,a,e
-234 -202 0
c T: 1101100111 -> [a,e]; T': 0010100111 -> [a]; P_i: b,c,d,a,e
Agent 0: b,c,d,a,e
Agent 1: a,e,d,b,c
Agent 2: e,c,d,b,a
Manipulated preferences of agent 0: a,c,b,d,e

-218 -218 0
c T: 1101100111 -> [a]; T': 1001000100 -> [e]; P_i: e,c,a,d,b
-232 -232 0
c T: 1101100111 -> [a,b,c,d]; T': 1001000100 -> [a,c,d,e]; P_i: e,c,a,d,b
Agent 0: e,c,a,d,b
Agent 1: d,a,e,b,c
Agent 2: d,a,e,b,c
Agent 3: d,e,b,c,a
Agent 4: c,e,b,d,a
Agent 5: b,c,a,e,d
Agent 6: b,a,c,e,d
Manipulated preferences of agent 0: b,a,c,d,e

-248 -338 0
c T: 1101100111 -> [a, b, c, d, e]; T': 1100100101 -> [b, c, e]; P_i: 1 > 4 >
2 > 3 > 0
Agent 0: 1 > 4 > 2 > 3 > 0
Agent 1: 2 > 3 > 0 > 4 > 1
Agent 2: 2 > 4 > 3 > 1 > 0
Agent 3: 0 > 4 > 3 > 1 > 2
Agent 4: 1 > 0 > 4 > 2 > 3
Manipulated preferences of agent 0:
1 > 2 > 0 > 4 > 3

-231 -202 0
c T: 1101100111 -> [b, c, d]; T': 1111111111 -> [e]; P_i: 0 > 4 > 1 > 2 > 3

```

```

-247 -202 0
c T: 11011001111 -> [b, c, d, e]; T': 11111111111 -> [e]; P_i: 0 > 4 > 1 > 2 > 3
Agent 0: 0 > 4 > 1 > 2 > 3
Agent 1: 3 > 1 > 2 > 0 > 4
Agent 2: 4 > 2 > 3 > 1 > 0
Manipulated preferences of agent 0:
4 > 0 > 3 > 2 > 1

-314 -314 0
c T: 11001011110 -> [c]; T': 1100100101 -> [e]; P_i: 1 > 3 > 4 > 0 > 2
Agent 0: 1 > 3 > 4 > 0 > 2
Agent 1: 4 > 3 > 1 > 2 > 0
Agent 2: 4 > 1 > 2 > 0 > 3
Agent 3: 2 > 0 > 3 > 4 > 1
Agent 4: 2 > 0 > 3 > 4 > 1
Manipulated preferences of agent 0:
1 > 2 > 0 > 4 > 3

c T: 11001011110 -> [c]; T': 11001101110 -> [b]; P_i: 1 > 2 > 3 > 4 > 0
Agent 0: 1 > 2 > 3 > 4 > 0
Agent 1: 0 > 3 > 4 > 2 > 1
Agent 2: 0 > 4 > 2 > 3 > 1
Agent 3: 4 > 1 > 2 > 0 > 3
Agent 4: 3 > 4 > 1 > 2 > 0
Manipulated preferences of agent 0:
3 > 1 > 2 > 0 > 4

-318 -318 0
c T: 11001011110 -> [d]; T': 1100100101 -> [b]; P_i: 1 > 3 > 4 > 0 > 2
c T: 11001011110 -> [d]; T': 11001101110 -> [a]; P_i: 1 > 2 > 4 > 0 > 3
Agent 0: 1 > 2 > 4 > 0 > 3
Agent 1: 3 > 4 > 1 > 2 > 0
Agent 2: 4 > 0 > 2 > 3 > 1
Agent 3: 2 > 0 > 3 > 4 > 1
Agent 4: 1 > 0 > 3 > 4 > 2
Manipulated preferences of agent 0:
3 > 1 > 2 > 0 > 4

-322 -322 0
c T: 11001011110 -> [c, d]; T': 11001101110 -> [a, b]; P_i: 1 > 4 > 0 > 2 > 3
Agent 0: 1 > 4 > 0 > 2 > 3
Agent 1: 2 > 0 > 3 > 4 > 1
Agent 2: 3 > 4 > 1 > 2 > 0
Manipulated preferences of agent 0:
1 > 0 > 3 > 4 > 2

-326 -326 0
c T: 11001011110 -> [e]; T': 11001101110 -> [d]; P_i: 1 > 2 > 3 > 4 > 0
Agent 0: 1 > 2 > 3 > 4 > 0
Agent 1: 0 > 3 > 4 > 2 > 1
Agent 2: 0 > 4 > 2 > 3 > 1
Agent 3: 4 > 1 > 2 > 0 > 3
Agent 4: 3 > 4 > 1 > 2 > 0
Manipulated preferences of agent 0:
3 > 1 > 2 > 0 > 4

```

```

-334 -202 0
c T: 1100101110 -> [d, e]; T': 1001111010 -> [d]; P_i: 2 > 0 > 3 > 4 > 1
Agent 0: 2 > 0 > 3 > 4 > 1
Agent 1: 1 > 4 > 0 > 2 > 3
Agent 2: 3 > 4 > 1 > 2 > 0
Manipulated preferences of agent 0:
3 > 1 > 0 > 4 > 2

202 0
Agent 0: 4 > 3 > 2 > 1 > 0

314 318 322 326 330 334 338 0
Agent 0: 2 > 0 > 3 > 4 > 1
Agent 1: 3 > 4 > 1 > 2 > 0
Agent 2: 4 > 1 > 2 > 0 > 3

```

References

- Barberà, S. (1977). Manipulation of social decision functions. *Journal of Economic Theory*, 15(2), 266–278.
- Barberà, S. (2010). Strategy-proof social choice. In Arrow, K. J., Sen, A. K., & Suzumura, K. (Eds.), *Handbook of Social Choice and Welfare*, Vol. 2, chap. 25, pp. 731–832. Elsevier.
- Biere, A. (2008). PicoSAT essentials. *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, 4, 75–79.
- Biere, A., Heule, M., van Maaren, H., & Walsh, T. (Eds.). (2009). *Handbook of Satisfiability*, Vol. 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.
- Blanchette, J. C., & Nipkow, T. (2010). Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In *Proceedings of the First International Conference on Interactive Theorem Proving*, pp. 131–146. Springer.
- Brandl, F., Brandt, F., Geist, C., & Hofbauer, J. (2015). Strategic abstention based on preference extensions: Positive results and computer-generated impossibilities. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 18–24. AAAI Press.
- Brandt, F. (2015). Set-monotonicity implies Kelly-strategyproofness. *Social Choice and Welfare*, 45(4), 793–804.
- Brandt, F., & Brill, M. (2011). Necessary and sufficient conditions for the strategyproofness of irresolute social choice functions. In *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pp. 136–142. ACM Press.
- Brandt, F., Brill, M., & Harrenstein, P. (2016a). Tournament solutions. In Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.), *Handbook of Computational Social Choice*, chap. 3. Cambridge University Press.
- Brandt, F., Geist, C., & Harrenstein, P. (2016b). A note on the McKelvey uncovered set and Pareto optimality. *Social Choice and Welfare*, 46(1), 81–91.

- Brandt, F., Geist, C., & Peters, D. (2016c). Optimal bounds for the no-show paradox via SAT solving. In *Proceedings of the 15th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 314–322. IFAAMAS.
- Brandt, F., Geist, C., & Seedig, H. G. (2014). Identifying k -majority digraphs via SAT solving. In *Proceedings of the 1st AAMAS Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE)*.
- Caminati, M. B., Kerber, M., Lange, C., & Rowat, C. (2015). Sound auction specification and implementation. In *Proceedings of the 16th ACM Conference on Economics and Computation (ACM-EC)*, pp. 547–564. ACM Press.
- Chatterjee, S., & Sen, A. (2014). Automated reasoning in social choice theory — some remarks. *Mathematics in Computer Science*, 8(1), 5–10.
- Ching, S., & Zhou, L. (2002). Multi-valued strategy-proof social choice rules. *Social Choice and Welfare*, 19(3), 569–580.
- Cinà, G., & Endriss, U. (2015). A syntactic proof of Arrow’s theorem in a modal logic of social choice functions. In *Proceedings of the 14th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 1009–1017. IFAAMAS.
- Conitzer, V., & Sandholm, T. (2002). Complexity of mechanism design. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 103–110.
- Drummond, J., Perrault, A., & Bacchus, F. (2015). SAT is an effective and complete method for solving stable matching problems with couples. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 518–525. AAAI Press.
- Duggan, J., & Schwartz, T. (2000). Strategic manipulability without resoluteness or shared beliefs: Gibbard-Satterthwaite generalized. *Social Choice and Welfare*, 17(1), 85–93.
- Erdamar, B., & Sanver, M. R. (2009). Choosers as extension axioms. *Theory and Decision*, 67(4), 375–384.
- Feldman, A. (1979). Manipulation and the Pareto rule. *Journal of Economic Theory*, 21, 473–482.
- Fishburn, P. C. (1972). Even-chance lotteries in social choice theory. *Theory and Decision*, 3(1), 18–40.
- Fishburn, P. C. (1977). Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3), 469–489.
- Fisher, D. C., & Reeves, R. B. (1995). Optimal strategies for random tournament games. *Linear Algebra and its Applications*, 217, 83–85.
- Fréchette, A., Newman, N., & Leyton-Brown, K. (2016). Solving the station repacking problem. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press.
- Gärdenfors, P. (1976). Manipulation of social choice functions. *Journal of Economic Theory*, 13(2), 217–228.

- Gärdenfors, P. (1979). On definitions of manipulation of social choice functions. In Laffont, J. J. (Ed.), *Aggregation and Revelation of Preferences*. North-Holland.
- Geist, C., & Endriss, U. (2011). Automated search for impossibility theorems in social choice theory: Ranking sets of objects. *Journal of Artificial Intelligence Research*, *40*, 143–174.
- Grandi, U., & Endriss, U. (2013). First-order logic formalisation of impossibility theorems in preference aggregation. *Journal of Philosophical Logic*, *42*(4), 595–618.
- Kelly, J. S. (1977). Strategy-proofness and social choice functions without single-valuedness. *Econometrica*, *45*(2), 439–446.
- Laslier, J.-F. (1997). *Tournament Solutions and Majority Voting*. Springer-Verlag.
- Liffiton, M. H., & Sakallah, K. A. (2008). Algorithms for computing minimal unsatisfiable subsets of constraints. *Journal of Automated Reasoning*, *40*(1), 1–33.
- McGarvey, D. C. (1953). A theorem on the construction of voting paradoxes. *Econometrica*, *21*(4), 608–610.
- McKay, B. D., & Piperno, A. (2013). Practical graph isomorphism, II. *Journal of Symbolic Computation*.
- Nehring, K. (2000). Monotonicity implies generalized strategy-proofness for correspondences. *Social Choice and Welfare*, *17*(2), 367–375.
- Nipkow, T. (2009). Social choice theory in HOL: Arrow and Gibbard-Satterthwaite. *Journal of Automated Reasoning*, *43*, 289–304.
- Sanver, M. R., & Zwicker, W. S. (2012). Monotonicity properties and their adaption to irresolute social choice rules. *Social Choice and Welfare*, *39*(2–3), 371–398.
- Sato, S. (2008). On strategy-proof social choice correspondences. *Social Choice and Welfare*, *31*, 331–343.
- Sato, S. (2013). A sufficient condition for the equivalence of strategy-proofness and non-manipulability by preferences adjacent to the sincere one. *Journal of Economic Theory*, *148*, 259–278.
- Scott, A., & Fey, M. (2012). The minimal covering set in large tournaments. *Social Choice and Welfare*, *38*(1), 1–9.
- Tang, P., & Lin, F. (2009). Computer-aided proofs of Arrow’s and other impossibility theorems. *Artificial Intelligence*, *173*(11), 1041–1053.
- Tang, P., & Lin, F. (2011). Discovering theorems in game theory: Two-person games with unique pure nash equilibrium payoffs. *Artificial Intelligence*, *175*(14–15), 2010–2020.
- Taylor, A. D. (2005). *Social Choice and the Mathematics of Manipulation*. Cambridge University Press.
- Tseitin, G. S. (1983). On the complexity of derivation in propositional calculus. In *Automation of Reasoning*, pp. 466–483. Springer.