CHAPTER 13

Computer-aided Methods for Social Choice Theory

Christian Geist and Dominik Peters

13.1 Introduction

The Four Color Theorem is a famous early example of a mathematical result that was proven with the help of computers. Recent advances in artificial intelligence, particularly in constraint solving, promise the possibility of significantly extending the range of theorems that could be provable with the help of computers. Examples of results of this type are a special case of the Erdős Discrepancy Conjecture (Konev and Lisitsa, 2014), and a solution to the Boolean Pythagorean Triples Problem (Heule et al., 2016). The proofs obtained in these two cases are only available in a computer-checkable format, and have sizes of 13 GB and 200 TB, respectively. Proofs like these do not have any hope of being human-checkable, and make the controversy about the proof of the Four Color Theorem pale in comparison. The computer-found proofs of results we discuss in this chapter, on the other hand, will have the striking property of being translatable to a human-readable version.

Social choice theory studies group decision making, where the preferences of several agents need to be aggregated into one joint decision. This field of study has three characteristics that suggest applying computer-aided reasoning to it: it uses the axiomatic method, it is concerned with combinatorial structures, and its main concepts can be defined based on rather elementary mathematical notions. Thus, Nipkow (2009) notes that "social choice theory turns out to be perfectly suitable for mechanical theorem proving." In this chapter, we will present a set of tools and methods first employed in papers by Tang and Lin (2009) and Geist and Endriss (2011) that, thanks to the aforementioned properties, will allow us to use computers to prove one of social choice theory's most celebrated type of result: *impossibility theorems*.

An impossibility theorem posits that there does not exist a preference aggregation procedure that satisfies a given set of desirable axioms, and that these axioms are therefore incompatible. For example, the Gibbard–Satterthwaite Theorem says that no voting rule is simultaneously non-dictatorial, onto, and strategyproof. (A *voting rule* takes as input a collection of preference rankings and selects a winning alternative.) How to prove such a theorem? If we begin by limiting the number of

alternatives and voters involved to a finite number, this task reduces to iterating through all possible voting rules and checking that none of them satisfies all axioms. However, this task invites an amazing combinatorial explosion, and a naïve search would be hopeless.

The main piece in our toolkit is a *SAT solver*, which can use our axioms to make logical inferences that make the search feasible. SAT solvers are computer programs that apply powerful reasoning strategies to decide whether a given propositional formula has a satisfying assignment or not. The last two decades have seen dramatic speedups in solving times, with regular 'SAT competitions' producing faster and faster solvers, despite the intimidating NP-completeness of the problem. Most solvers are freely available for use by researchers.

Here is the proof technique in a nutshell: Fix some finite number n of voters and m of alternatives, say $n = m = 3$. Produce a formula of propositional logic saying "there exists a voting rule for n voters and m alternatives satisfying such and such axioms" and decide its satisfiability using any state-of-the-art SAT solver. If, on the one hand, the formula is satisfiable, we can extract a voting rule that satisfies the desired axioms. If, on the other hand, the formula is unsatisfiable, we have the beginnings of an impossibility theorem. Usually, we now want two more things: find a *proof* of the unsatisfiability (human-readable if possible), and to extend the result to larger n and m. To achieve the first goal, we will present an exciting method using *minimal unsatisfiable sets*. For the second goal, we can often prove a relatively straightforward *induction step*, establishing an impossibility for arbitrarily large n and m.

This technique has been successfully applied by Tang and Lin (2008, 2009) to find new proofs of Arrow's and of the Gibbard–Satterthwaite Theorem, by Geist and Endriss (2011) to find new impossibilites in the space of set extensions, by Brandt and Geist (2016) and Brandl et al. (2015) to study strategic properties of tournament solutions, by Brandt et al. (2017) to study the no-show paradox, and by Brandl et al. (2016) to study probabilistic social choice. Many other applications of the technique are imaginable.

Related Work. Using logic solvers has proven to be useful for other problems in economics, too. Examples are the work by Fréchette et al. (2016), in which SAT solvers are used for the development and execution of the FCC's reverse spectrum auction, and recent results by Drummond et al. (2015), who solve stable matching problems via SAT solving. Closely related to our approach is an article by Tang and Lin (2011), who apply SAT solving to identify classes of two-player games with unique pure Nash equilibrium payoffs. In another recent paper, Caminati et al. (2015) verified combinatorial Vickrey auctions via higher-order theorem provers.

In some respects, our approach is similar to *automated mechanism design* (see, e.g., Conitzer and Sandholm, 2002), where desirable properties are also encoded as constraints, but mechanisms are computed to fit specific problem instances (rather than being applicable generally). In a similar spirit, Mennle and Seuken (2015) run linear programs in order to compute optimal (randomized) choice mechanisms that satisfy approximate versions of strategyproofness and efficiency.

A related line of work is directed towards formalizing and verifying *existing*

results of social choice theory, such as Arrow's Impossibility Theorem, through logical formalizations (see, e.g., Nipkow, 2009; Grandi and Endriss, 2013; Ciná and Endriss, 2016).

The method we are going to survey in this chapter has also been discussed in articles by Chatterjee and Sen (2014) and Kerber et al. (2016).

Chapter Outline. This chapter comes in two main parts. First, in Section 13.2 we go through an application of the method step-by-step using a simple toy impossibility. Second, in Section 13.3, we give a detailed survey of several papers that have applied the method, emphasising key extensions to the basic technique as described in Section 13.2. We close by discussing the potential and the limits of the method, and sketch some possibilities for future work.

13.2 Case Study: Strategyproofness and the Majority Criterion

In this section, we showcase the method by going through a toy example in detail. We will use SAT solvers to find a human-readable proof of the following result:

Theorem 13.1. *For $n = 3$ voters and $m \geqslant 3$ alternatives, no (resolute) voting rule satisfies both strategyproofness and the majority criterion.*

Let $N = \{1, 2, 3\}$ be a set of 3 voters, let $A = \{a, b, c\}$ be the set of alternatives, let $\mathcal{L}(A)$ be the set of linear orders over A, and let $\mathcal{R} = \mathcal{L}(A)^N$ be the set of profiles over A. Thus, a profile is a function assigning a preference ordering to each voter.

A resolute voting rule $f : \mathcal{R} \to A$ is a function mapping every preference profile of linear orders to a winning alternative.

The voting rule f satisfies the *majority criterion* if whenever R is a profile so that a strict majority of voters ranks some alternative $x \in A$ in top position, then $f(R) = x$.[1] For convenience, for each $x \in A$, we write $\mathcal{M}(x) \subseteq \mathcal{R}$ for the set of profiles where x is ranked top by a majority.

We say that f is *manipulable* if there is a voter $i \in N$ and two profiles $R, R' \in \mathcal{R}$ such that $R(j) = R'(j)$ for every voter $j \in N \setminus \{i\}$, but $f(R') \succ_i f(R)$, where $\succ_i = R(i)$ is i's vote in R. Thus, voter i can achieve a better outcome by misreporting her preferences as $\succ'_i = R'(i)$. Conversely, f is *strategyproof* if f is not manipulable.

Note that the Gibbard–Satterthwaite Theorem immediately implies our Theorem 13.1: if a voting rule satisfies the majority criterion, then it is also onto and non-dictatorial. Thus, proving Theorem 13.1 is not a breakthrough result; however, we hope that this section illustrates the SAT approach.

Step 1: Encoding into SAT. We start by constructing a formula φ of propositional logic that is satisfiable if and only if there exists a resolute voting rule f for $n = 3$ voters and $m = 3$ alternatives satisfying both strategyproofness and

[1] Thus, the majority criterion is a much weaker axiom than Condorcet-consistency. The Borda count is a notable example of a rule which fails the majority criterion.

the majority criterion. So that the formula can be processed by standard SAT solvers, we will ensure that our encoding produces a formula in *conjunctive normal form* (CNF). Thus, φ will be a conjunction of *clauses*. Recall that a clause is a disjunction of literals, and a literal is either a variable or its negation. In our example, it will be very natural to phrase our problem as a CNF formula. For some other settings or axioms, we might need to transform a non-CNF formula into CNF first. This is best done using Tseitin transformations (Tseitin, 1983; see Kroening and Strichman, 2016, p. 12–14, for examples).

The propositional variables used in φ will describe the voting rule f explicitly, i.e., they will specify the output of f for every possible profile with $n = m = 3$. Precisely, we will use one variable $v_{R,x}$ for each such profile $R \in \mathcal{R}$ and each $x \in A$. The intended interpretation is that

$$f(R) = x \iff v_{R,x} \text{ is true.}$$

With this interpretation, any satisfying assignment of φ will give rise to a voting rule satisfying our axioms. Thus, Theorem 13.1 will be true (for $n = m = 3$) if and only if φ is unsatisfiable.

We now describe the formula φ as the conjunction of several subformulas. We start out by formalising the requirement that f be a function, that is, $f(R)$ needs to correspond to exactly one alternative. This requirement can be broken down into two statements: that there be *at least* one such alternative, and *at most* one such alternative. The first of these is easy to encode:

$$\varphi_{\text{at least one}} \equiv \bigwedge_{R \in \mathcal{R}} (v_{R,a} \vee v_{R,b} \vee v_{R,c})\,.$$

The second part can be phrased in CNF by requiring that $f(R)$ does not correspond to two distinct alternatives:

$$\varphi_{\text{resolute}} \equiv \bigwedge_{R \in \mathcal{R}} \bigwedge_{\substack{x,y \in A \\ x \neq y}} (\neg v_{R,x} \vee \neg v_{R,y}).$$

Next, we encode our axioms. The majority criterion is relatively easy to put into logic; we just identify all profiles R for which the majority criterion implies a restriction on f. Recall that $\mathcal{M}(x)$ is the set of profiles in which a majority of voters puts x in top position.

$$\varphi_{\text{majority}} \equiv \bigwedge_{x \in A} \bigwedge_{R \in \mathcal{M}(x)} (v_{R,x}).$$

Strategyproofness is an axiom relating the output of f at two different profiles. The encoding below interprets strategyproofness as "if $f(R) = y$, and voter i can change profile R into R' by misrepresenting her preferences, then $f(R')$ cannot be better for i than y".

$$\varphi_{\text{strategyproof}} \equiv \bigwedge_{i \in N} \bigwedge_{R \in \mathcal{R}} \bigwedge_{\substack{R' \in \mathcal{R} \\ R'(j) = R(j) \\ \forall j \neq i}} \bigwedge_{\substack{x,y \in A \\ x \succ_i y}} (v_{R,y} \to \neg v_{R',x}).$$

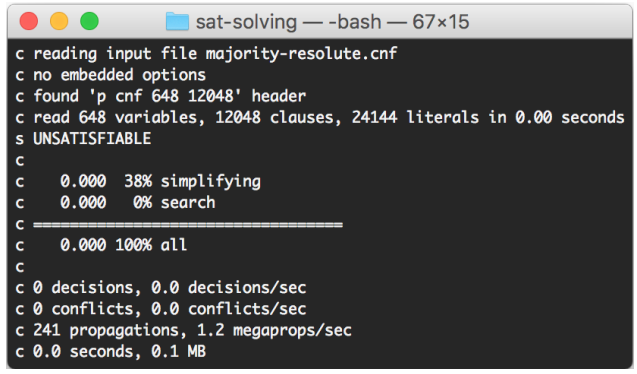

Figure 13.1: Command-line output of the `lingeling` SAT solver when run on the formula produced for our example. The line `s UNSATISFIABLE` states the result.

(Implications "$a \to b$" can be rewritten as the disjunction $\neg a \vee b$.) Putting all the subformulas together, we finally obtain

$$\varphi = \varphi_{\text{at least one}} \wedge \varphi_{\text{resolute}} \wedge \varphi_{\text{majority}} \wedge \varphi_{\text{strategyproof}}.$$

Step 2: SAT Solving. Now that we have an encoding of our problem into CNF, we can pass it on to a SAT solver to decide whether φ is satisfiable. For this we will need to choose a specific solver, and write φ in a file format that the solver can understand. A common source for finding powerful solvers is the series of "SAT competitions" (`satcompetition.org`), where solvers by different researchers are compared on a benchmark set. Two good choices at the time of writing are `lingeling` (developed by Biere, 2013) and `glucose` (developed by Audemard and Simon, 2009).

All common SAT solvers accept input formulas as text files in the DIMACS format. To produce a CNF formula in this format, identify the propositional variables using integers $1, 2, \ldots, v$. Each clause of the formula is represented by a line in the text file; for example, the clause $1 \vee \neg 2 \vee 3$ is represented by the line

```
1 -2 3 0
```

Literals are separated by spaces and lines are terminated with a `0` and a newline character. The first line of a DIMACS file is a header which contains the number of variables and clauses used in the formula. For example, the formula we produced to encode our problem here contains 648 variables and 12,048 clauses, and so the associated file would start with the line

```
p cnf 648 12048
```

The DIMACS file should usually be produced using a script which keeps track of the mapping from variables to integers, so that a possible satisfying assignment can be translated back into an actual model.

When running our chosen SAT solver on the formula produced, we obtain a report that φ is unsatisfiable in much less than 1 second[2] (as in the screenshot in Figure 13.1).

Step 3: MUS Extraction. Still, we would like to know a *reason* for the unsatisfiability, and preferably a human-readable proof. For these purposes it will be useful to know where the unsatisfiability comes from, and which profiles and constraints were responsible for it. The relevant tool for this goal is a *minimal unsatisfiable set (MUS)*, which is a subset of the clauses of φ that is already unsatisfiable, and is minimally so, in the sense that removing any of the clauses results in a satisfiable formula. Thus, each of the clauses in an MUS encodes a 'proof step' that cannot be skipped. The fewer clauses an MUS contains, the easier it is to understand, and so we are hoping to find an MUS of small cardinality.

The number of available tools for extracting an MUS is substantially smaller than the number of SAT solvers. For our purposes the currently best tool available is `MUSer2` by Belov and Marques-Silva (2012), which internally uses the solver `glucose` as a SAT oracle. `MUSer2` takes an unsatisfiable DIMACS file as input and returns an MUS. `MUSer2` also supports the computation of *group MUSes*: in this setting, the clauses of a CNF formula are partitioned into groups, and we are looking for an inclusion-minimal set of groups that are unsatisfiable. In our experience, it can be useful to group together the clauses referring to a single profile, as this can lead to smaller MUSes that are easier to interpret. A group-CNF formula is specified by a DIMACS file with clause lines like `{6} 1 -2 3 0`, where `6` is the number of the group that corresponds to this clause, and has header line `p gcnf 648 12048 100`, where `100` is the number of groups used. Clauses in group `0` are never removed by `MUSer2`.

Another useful tool is MARCO (Liffiton et al., 2015), which enumerates *all* MUSes of a given (group-)CNF formula. This can be helpful to find smaller MUSes, or ones that have a simpler structure.

The MUS obtained in our example contains just 9 clauses, which refer to a total of 5 profiles, which we label $R_\alpha, R_\beta, R_\gamma, R_\delta, R_\epsilon$ and show in Figure 13.2. In particular, the MUS contains 2 clauses from $\varphi_{\text{at least one}}$ for R_α and R_γ, 3 clauses from $\varphi_{\text{majority}}$ for R_β, R_δ, and R_ϵ, and 4 clauses from $\varphi_{\text{strategyproof}}$ for manipulations from R_α to R_γ, from R_α to R_β, from R_γ to R_δ, and from R_γ to R_ϵ. Interestingly, the MUS does not contain any clauses from $\varphi_{\text{resolute}}$.

Step 4: Interpreting the MUS. Now we will translate the MUS obtained in the previous step into a human-readable proof. The MUS we have obtained can be displayed in a graphical fashion, as in the right-hand half of Figure 13.2. Producing such "proof diagrams" can make it much easier to produce a human-readable proof. To see how this diagram can be obtained, it is useful to distinguish between *intra-profile* and *inter-profile* axioms (cf. Fishburn, 1973). An intra-profile axiom refers to the allowed voting outcomes at a *single* profile; examples are Pareto optimality, the majority criterion, and Condorcet-consistency. Any clauses

[2] For our example problem, the strategy of *unit propagation* turns out to be enough to solve the formula, but this is not the case for all problems of this type.

	voter 1	voter 2	voter 3
R_α	$a \succ b \succ c$	$b \succ a \succ c$	$c \succ a \succ b$
R_β	$a \succ b \succ c$	$b \succ a \succ c$	$a \succ b \succ c$
R_γ	$a \succ b \succ c$	$b \succ c \succ a$	$c \succ a \succ b$
R_δ	$a \succ b \succ c$	$c \succ a \succ b$	$c \succ a \succ b$
R_ϵ	$b \succ a \succ c$	$b \succ c \succ a$	$c \succ a \succ b$

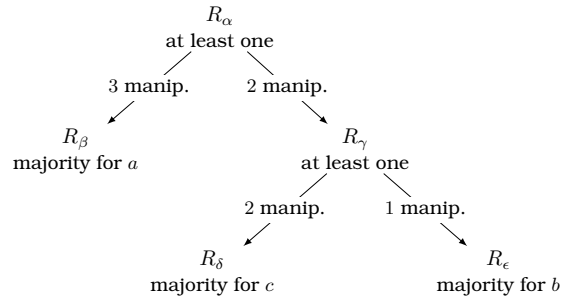

Figure 13.2: The 5 profiles used in the MUS, and a graphical representation of the clauses in the MUS.

in the MUS referring to an intra-profile axiom can be written next to the node of the diagram representing the profile under consideration. An inter-profile axiom *connects* the voting outcomes at multiple profiles; examples are strategyproofness, participation, anonymity and neutrality. Note that these four examples can be phrased so as to refer to *exactly two* profiles. As such, they can be displayed as a (possibly directed) edge connecting two profile-nodes in the diagram. In our example, directed edges correspond to clauses specifying that a certain manipulation is not successful. Proof diagrams of this sort, first used by Brandt et al. (2017), can serve as a unifying 'language' of impossibility proofs. Note, however, that it is unclear how to represent inter-profile axioms that refer to more than two profiles, such as non-dictatorship or non-imposition.

We will now translate the 'proof' shown in Figure 13.2 into English. In reading the following proof, it is useful to refer to the diagram to see how the two objects correspond to each other.

Theorem 13.2 (Base case). *For $m = n = 3$, there is no voting rule that satisfies strategyproofness and the majority criterion.*

Proof. Suppose f is a voting rule satisfying both axioms. We proceed in a bottom-up fashion, establishing which value f must take for each profile, and find that there is no possible value $f(R_\alpha)$ for the root node, contradiction.

Consider the profile R_γ, where $f(R_\gamma)$ needs to take some value. If $f(R_\gamma) = a$, then voter 2 can manipulate to obtain profile R_δ. By the majority criterion, $f(R_\delta) = c$, and so we have $f(R_\delta) \succ_2 f(R_\gamma)$, and this was a successful manipulation, contradicting strategyproofness of f. Hence $f(R_\gamma) \neq a$.

If $f(R_\gamma) = c$, then voter 1 can manipulate to obtain profile R_ϵ. By the majority criterion, $f(R_\epsilon) = b$. Thus $f(R_\epsilon) \succ_1 f(R_\gamma)$, and this was a successful manipulation, contradicting strategyproofness of f. Hence $f(R_\gamma) \neq c$. Hence $f(R_\gamma) = b$.

By the majority criterion, $f(R_\beta) = a$. Now consider profile R_α. In case $f(R_\alpha) = b$, then voter 3 can manipulate to obtain profile R_β. Since $f(R_\beta) = a$, this would be a successful manipulation for voter 3. Hence $f(R_\alpha) \neq b$.

Thus, $f(R_\alpha) \in \{a, c\}$. But voter 2 can manipulate at R_α to obtain profile R_γ. We have seen that $f(R_\gamma) = b$. Hence $f(R_\gamma) \succ_2 f(R_\alpha)$ according to voter 2's preferences in R_α, contradicting strategyproofness of f. Hence such an f cannot exist. □

Step 5: Induction Step. We have found a proof of the finite fact that for $n = m = 3$, no voting rule can satisfy the axioms of strategyproofness and of the majority criterion. This alone is perhaps not completely satisfying, since we would like to say something about larger values of n and m. Often, it is possible to prove a *reduction argument* which consists of statements of the form

- If there is a voting rule satisfying a given set of axioms for $n+1$ voters and m alternatives, then there is also a voting rule satisfying these axioms for n voters and m alternatives.
- If there is a voting rule satisfying a given set of axioms for n voters and $m+1$ alternatives, then there is also a voting rule satisfying these axioms for n voters and m alternatives.

Viewed contrapositively, these statements are *induction steps*. If we can prove these statements, given our proof of the base case above, it immediately follows that the impossibility result also holds for all larger values of n and m. Sometimes it can be difficult to establish this induction step; in such cases it may be possible to manually extend the proof obtained for the base case to go through for larger n or m. In some settings, the induction step may turn out to be false (for example, an impossibility could only hold in profiles with an odd number of voters). For our example, it is easy to prove an induction step for m, but we do not know of a way to prove the induction step for n (except, of course, by cheating and appealing to the Gibbard–Satterthwaite Theorem directly).

Lemma 13.3. *Let $m \geqslant 3$ and $n = 3$. If f is a resolute voting rule satisfying strategyproofness and the majority criterion for $m+1$ alternatives, then there exists a voting rule f' for m alternatives with the same properties.*

Proof. Let f be defined for the alternative set $A \cup \{x\}$, where $|A| = m$. For every profile R with n voters and on alternative set A, define the profile R^{+x} derived from R by putting alternative x at the bottom of each preference order. Thus, for each voter i and all $a, b \in A$, we have $a \succsim_i^{+x} b$ if and only if $a \succsim_i b$, and we have $a \succsim_i^{+x} x$ for each $a \in A$. Then define the rule f' on alternative set A by

$$f'(R) := f(R^{+x}).$$

Clearly, f' satisfies the majority criterion, for if x is the majority winner in R, then x is also majority winner in R^{+x}, so $f'(R) = f(R^{+x}) = x$, since f satisfies the majority criterion. Also, f' satisfies strategyproofness, because any successful manipulation of f' from R to R' is a successful manipulation of f from R^{+x} to R'^{+x}, which would contradict strategyproofness of f. □

The technique of adding new alternatives to the bottom of the profile is a standard tool for reducing the number of alternatives. Often-used moves for reducing the number of voters, on the other hand, include cloning a specific voter, adding an all-indifferent voter, or adding two voters with completely reversed preferences.

Having established both base case and induction step, this concludes the proof of our modest impossibility theorem.

13.3 Applications and Advanced Techniques

We will now survey several papers that have used this style of technique to prove much more complex results, and discuss ways in which they deviate from the recipe.

13.3.1 Arrow's and the Gibbard–Satterthwaite Theorem

The two most famous impossibility theorems in social choice theory are Arrows' Impossibility Theorem and the Gibbard–Satterthwaite Theorem. Tang and Lin (2008, 2009) used these theorems to introduce the idea of proving impossibilities in social choice by induction on n and m, and by verifying the base case using a SAT or constraint-programming solver. It should be clear how to adapt the method of Section 13.2 to prove the base case of the Gibbard–Satterthwaite Theorem; to encode non-imposition one could use $\bigwedge_{x \in A} \bigvee_R v_{R,x}$, and to encode non-dictatorship one could use $\bigwedge_{i \in N} \bigvee_R \neg v_{R,\mathrm{top}(R,i)}$, where $\mathrm{top}(R,i)$ denotes the most-preferred alternative of voter i in profile R. Note that both of these axioms use very long clauses, which can be tough for solvers to use. Similar ideas can be used to encode the base case of Arrow's Impossibility Theorem; for this, one would probably introduce a variable $v_{R,\succsim}$ for each profile R and each possible output relation $\succsim$.

While establishing the base case for both of these theorems is straightforward, establishing the induction steps is rather involved. This is in contrast to the results we will see below, where establishing the base cases is the main technical difficulty. This is because the number of variables in the encoding grows exponentially with n and m, and the results below need base cases with $m > 3$.

13.3.2 Irresolute SCFs and Fishburn's Set Extension

In this section we will see two modifications to the technique we saw in Section 13.2: here, we will consider voting rules that are *majoritarian* and *set-valued*. Both of these changes have impacts on the encoding technique.

A *set-valued* (or *irresolute*) voting rule assigns to every preference profile a non-empty subset of A; the usual interpretation is that the alternatives returned are tied for winner, and that some external tie-breaking mechanism will later be applied. This approach raises some problems for defining axioms such as strategyproofness; to extend the definition of the resolute case, we need to extend a voter's preference order to a preference order over *sets* of alternatives. Several different ways to do this (so-called *set extensions*) have been proposed. In this section, we will focus on Fishburn's set extension.[3] Suppose i is a voter with preference relation $\succsim_i$, and suppose i expects the ties in the voting rule to be broken according to some linear tie-breaking order; however, i does not know which order will be used. Now, if $X, Y \subseteq A$ are non-empty subsets of alternatives, set X is weakly preferred to Y according to the Fishburn extension (written $X \succsim_i^F Y$) if the tie-broken outcome from X is guaranteed to be weakly better

[3] Some authors refer to this set extension as Gärdenfors' extension. We follow the terminology of the survey by Gärdenfors (1979), who uses his own name for a different set extension.

(according to $\succsim_i$) than the outcome from Y, no matter the tie-breaking order used. For a more explicit definition, see Brandt and Geist (2016). Note that the relation $\succsim_i^F$ is incomplete: not all pairs of sets can be compared using the criterion given.

In a similar way to the resolute case, we say that an irresolute rule f is *Fishburn-manipulable* if for some voter $i \in N$ and for two profiles R and R' that differ only in i's vote, we have $f(R') \succ_i^F f(R)$, where $\succ_i$ is i's vote in profile R. Then f is *Fishburn-strategyproof* if it is not Fishburn-manipulable. Now let $R - i = R|_{N\setminus\{i\}}$ denote the profile obtained from R by removing voter i. We say that an irresolute rule f (defined over profiles with variable electorates) satisfies *Fisburn-participation* if there is no profile R and voter i such that $f(R-i) \succ_i^F f(R)$. Thus, voters never strictly regret having voted.

Further, f is said to be *majoritarian* if it is neutral and selects the same set of alternatives for any two profiles with the same majority relation, and it is *Pareto optimal* if whenever x is a Pareto dominated alternative in R, then $x \notin f(R)$.

Finally, we can state the two impossibilities of this section.

Theorem 13.4 (Brandt and Geist, 2016). *There is no majoritarian and Pareto optimal set-valued voting rule that satisfies Fishburn-strategyproofness if* $m \geqslant 5$ *and* $n \geqslant 7$*.*

Theorem 13.5 (Brandl et al., 2015). *There is no majoritarian and Pareto optimal set-valued voting rule that satisfies Fishburn-participation if* $m \geqslant 4$ *and* $n \geqslant 6$*.*

Both of these results were proved using the method presented in Section 13.2. Let us sketch how the encoding can be adapted to the new setting. In order to encode the irresoluteness of the voting rules, it is useful to choose a variable $v_{R,X}$ for every profile R and every non-empty $X \subseteq A$ indicating that $f(R) = X$. While this might seem like a wasteful encoding, requiring $2^m - 1$ variables for each profile, it makes it much easier to encode Fishburn's set extension, because we can evaluate the relation $\succsim^F$ at encoding time, rather than having to translate its definition into propositional logic.[4] To encode that f should be majoritarian, one could use clauses enforcing an if-and-only-if relation between the outputs of two profiles with the same majority relation. However, a much more efficient possibility exists: for each possible majority relation (represented by a tournament T), introduce variables $v_{T,X}$. Since there are vastly fewer tournaments than there are profiles, this uses many fewer variables, and clearly, this is enough to define any majoritarian voting rule. However, with this choice of encoding, one needs to take care to encode Fishburn-strategyproofness and neutrality only in terms of tournaments; see Theorem 1 and Lemma 1 of Brandt and Geist (2016).

13.3.3 The No-Show Paradox and Incremental Proof Discovery

A stunning result of Moulin (1988) establishes that no Condorcet extension satisfies participation, an axiom that requires that no voter can be worse off by participating and voting honestly. Moulin's proof requires 4 alternatives and 25 voters to go through. Since the maximin rule (with some fixed tie-breaking rule) is

[4] If one uses the optimistic or pessimistic preference extension (like in Brandt et al., 2017, Section 7), using variables $v_{R,x}$ for $x \in X$ can often be made to work directly.

a Condorcet extension satisfying participation if $m = 3$ (Moulin, 1988), we see that 4 alternatives are required for the result to hold. However, it seems unlikely that exactly 25 voters are required. We will use SAT solvers to find an optimal bound.

Formally, in this section we consider voting rules that are defined for *variable electorates*, i.e., that are defined for profiles that contain different numbers of voters. Given a profile R of linear orders, an alternative $a \in A$ is called a *Condorcet winner* if for every other alternative $b \in A \setminus \{a\}$, there is a strict majority of voters who prefer a to b. A voting rule is a *Condorcet extension* if it selects the Condorcet winner for all profiles that have one. As in the previous section, we say that a voting rule f satisfies *participation* if for all profiles R, and for all voters i who participate in R, we have $f(R) \succsim_i f(R - i)$. Thus every voter weakly prefers submitting their truthful preference order to abstaining. Again, $R - i = R|_{N\setminus\{i\}}$ is the profile obtained from R by removing voter i.

Theorem 13.6 (Brandt et al., 2017). *While there is no Condorcet extension that satisfies participation for $m \geqslant 4$ and $n \geqslant 12$, there exists such a voting rule for $m = 4$ and $n \leqslant 11$.*

In principle, it should be no mystery how to achieve such a result using the technique of Section 13.2. Inconveniently, there are $4!^{12} \approx 10^{16}$ profiles with $m = 4$ and $n = 12$, and it is impractical to enumerate them all in order to write down (let alone solve) the SAT formula we would produce.

One could try something similar to the approach of Section 13.3.2 and only consider majoritarian voting rules. In fact, if we restrict attention to *pairwise* (C2) voting rules (those that depend only on the *weighted* majority relation), then one gets a positive result for $n = 11$ and a negative one for $n = 12$. This result for $n = 11$ is useful as it implies the second part of Theorem 13.6; the result for $n = 12$, however, is weaker than desired because it uses the additional "axiom" that the voting rule is pairwise. However, obtaining this proof with an additional axiom could still be useful. Brandt et al. (2017) propose using what they call *incremental proof discovery*: using proofs of weaker statements to make educated guesses about a restricted domain over which to look for an impossibility result. In particular, they noticed that the impossibility proof for $n = 12$ using pairwiseness had some interesting structure. The profiles in the proof did contain all $4! = 24$ possible preference orders, instead using only 10 different orders. They then produced a formula including only profiles that were built up using only these 10 orders; a much smaller formula. This formula turned out to be already unsatisfiable, establishing the first part of Theorem 13.6.

Brandt et al. (2017) then used information gleaned from the proof for $n = 12$ to search for impossibility results to look for impossibilities for set-valued voting rules (with participation defined using the *optimistic* and *pessimistic* set extensions) and found optimal bounds of $n = 17$ and $n = 14$ for impossibility results in this setting. Later, Peters (2017) used similar techniques to find such results for Condorcet extensions satisfying *half-way monotonicity*, which is a weaker condition than participation.

13.3.4 Probabilistic Voting Rules

Let $\Delta(A)$ be the set of lotteries (probability distributions) over A. A *probabilistic voting rule* (also known as a *social decision scheme*) assigns a lottery to each preference profile; for example, the voting outcome might be a fair coin toss between a and b. As usual, such a voting rule is anonymous and neutral if it is invariant under renaming voters and alternatives, respectively. For defining other axioms, it is useful to have a way of comparing different lotteries in terms of their desirability to a given voter. Here, we will use the notion of stochastic dominance: If a voter i has preferences $\succsim_i$, and $p, q \in \Delta(A)$ are lotteries, we say that $p \succsim_i^{SD} q$ if

$$\sum_{y \succsim_i x} p(y) \geqslant \sum_{y \succsim_i x} q(y) \quad \text{for all } x \in A.$$

The main appeal of stochastic dominance stems from the following equivalence: $p \succsim^{SD} q$ if and only if p yields at least as much von-Neumann-Morgenstern utility as q under *any* utility function that is consistent with the ordinal preferences $\succsim$. We can now say that a probabilistic voting rule f is *SD-strategyproof* if we do not have $f(R') \succ_i^{SD} f(R)$ for any profiles R and R' that differ only in voter $i \in N$, where $\succsim_i$ is i's preference according to R. Further, we say that f is *SD-efficient* if there never exists a lottery $p \neq f(R)$ such that $p \succsim_i^{SD} f(R)$ for all voters $i \in N$ and $p \succ_i^{SD} f(R)$ for some voter $i \in N$.

It turns out that these two axioms are incompatible in the presence of symmetry axioms.

Theorem 13.7 (Brandl et al., 2016). *If $m \geqslant 4$ and $n \geqslant 4$, and allowing weak orders with indifferences as individual preferences, there is no anonymous and neutral probabilistic voting rule that satisfies SD-efficiency and SD-strategyproofness.*

This result is notable because it implies several other previously found impossibility results in probabilistic social choice (see Chapter 1). Since this result appears in a chapter on computer-aided methods, it is unsurprising that it, too, was obtained using solving techniques. However, this may seem puzzling: even fixing $n = m = 4$, the search space of probabilistic voting rules is infinite. This suggests that we will need a different solving technique, which can deal with real-valued (rather than Boolean) variables. Integer Programming comes to mind, and indeed one can encode the axioms of Theorem 13.7 using Integer Programming.

Another option in this context, though, are *SMT solvers* ("satisfiability modulo theories"). These solvers are very flexible and are used mainly in software verification. When we use linear arithmetic as our underlying theory, we can think of the input formula given to the SMT solver as a propositional formula whose atoms are linear (in)equalities of real-valued variables. Encoding our axioms into this language is relatively straightforward, though SD-efficiency requires some further analysis (see Brandl et al. (2016) for details). We have found that SMT solvers frequently solve problems like the ones discussed here faster than commercial Integer Programming solvers. This may be because branch-and-cut is less appropriate than conflict-driven approaches for our problems.

Unfortunately, the infinite search space involved seems to make this problem significantly tougher, and Brandl et al. (2016) found it prohibitive to search over

the entire space of about 1 million profiles. Thus, they looked for a similar reduction in the size of the search space as was successful for participation (Section 13.3.3). They noticed that many impossibility results take the form of starting at some initial profile R and then considering "nearby" profiles that can be obtained from R through few manipulations (i.e., few voters changing their reported preferences). They identified a promising profile R in which the popular probabilistic rule *Random Serial Dictatorship* returned an SD-inefficient lottery, and then generated a "ball" of about $10,000$ profiles around R which could be reached by at most 4 successive manipulations. They also only considered "small" manipulations, where the reported preference was close to the truthful one (according to the Kendall-tau distance). This domain was small enough for the solver to terminate, and to yield an unsatisfiability result. Many SMT solvers allow for the extraction of minimal unsatisfiable sets, which allow obtaining a human-readable proof, though this is much more involved in this case (see Eberl, 2016).

13.3.5 Other Applications

Set Extensions. There is a well-developed literature about how to extend a preference order over alternatives to a preference order over *sets* of alternatives; see Barberà et al. (2004) for a survey. Much of the work in this area focuses on axiomatic characterisations and impossibility results. Geist and Endriss (2011) introduce a logic for capturing many of the axioms used in the field, and prove a universal induction step for all (conjunctions of) axioms that can be represented as an "existentially set-guarded" formula in this logic. They then considered 20 of such axioms, and used a solver to check base cases for all combinations of these axioms, finding a total of 84 (axiom-minimal) impossibilities, several of which were not previously known.

Proof Verification. One way to gain more confidence in the correctness of computer-generated (and also human-generated) proofs is to formally check them. One tool to do so is Isabelle/HOL (Nipkow et al., 2002), which is a generic *interactive theorem prover*, where *interactive* indicates that the process of proof discovery is guided by a human operator who indicates a sequence of steps to follow, with most gaps filled automatically by the theorem prover. For examples of this applied to social choice, see Nipkow (2009) and Eberl (2016). In particular, the latter paper verifies the result from Section 13.3.4.

Solving-based Algorithms. There are a few examples of algorithms for computational problems in social choice theory that are powered by SAT solving. For instance, mostly by computing counterexamples, Brandt et al. (2016) explore the boundaries of the connection between the McKelvey uncovered set and the notion of Pareto optimality, Bachmeier et al. (2016) improve our understanding of the notions of k-majority digraphs, and Geist (2014) computes minimal preference profiles with Condorcet dimension 3. Interestingly, using solvers outperforms existing tailor-made algorithms in many applications.

13.4 Discussion and Future Work

New Insights. A perhaps surprising feature of computer-aided proofs is that they may give new and unexpected insights into the problems considered. For example, in studying the no-show paradox, Brandt et al. (2017) found proofs that exploit symmetries that were not present in Moulin's (1988) original proof. A related feature of the computer-aided methods discussed in this chapter are that they allow searching through various related conjectures. For example, it is easy to replace axioms by weaker versions to see whether the impossibility still holds. Brandt and Geist (2016) extensively used this technique to find several weakenings of Fishburn-strategyproofness that sill produce an impossibility. Geist and Endriss (2011) used this technique to find an exhaustive list of impossibility results in their domain of set extensions.

Better Usability. In their current form, applying solving methodologies to social choice theory remains a task for expert users with programming skills. While this does not impact the overall power of the approach, it limits the degree to which it can be broadly used by any researcher in social choice. Still, one may hope for the development of user-friendly tools that help formalizing concepts of social choice theory in the languages of solvers.

However, in our experience, the design of efficient encodings has to follow the requirements of—and needs to be optimized for—the concrete problem. This follows the observation that for general proof assistants with highly expressive input languages, such as Isabelle, many problems can be easily and intuitively formalized, but the ability of these systems to discover *new* results is rather limited due to the high complexity of the general problem.

Yet, some basic toolsets to assist *expert users* when formalizing concepts from social choice are certainly desirable and should be achievable based on the similarities of existing approaches. It remains an interesting question to which extent such tools can take the role of an automatic proof assistant which allows researchers to quickly test hypotheses on small domains without giving up too much generality and efficiency.

Limitations of the Approach. The vision that Tang (2010) had when he invented the basis for the methods presented here was computer-aided *theorem discovery*, which in his words includes two aspects: "to come up with reasonable conjectures automatically" and "to prove or disprove the conjectures automatically". While these targets turned out to be achievable in the domain of set extensions (Geist and Endriss, 2011), for more complex settings, we will usually need to come up with reasonable conjectures manually, and often the proving process cannot reasonably be described as "automatic". Based on this experience, we believe that the key for successful application of computer-aided methods will be a close collaboration between subject matter experts (who formulate the questions and provide theoretical tools) and experts on the method (who answer the questions with the help of machine support). This enables faster testing of conjectures, and also helps to explore similar statements as well as limits of the hypotheses.

When applied interactively, such collaboration might even guide the search for new results in cases where the conjectures are not clearly formulated yet, for instance by quickly providing counterexamples to some ideas.

Regarding the types of theorems that can be proven with the presented approach, there neither is an obvious classification nor are there strict limiting factors that are easily recognizable. An intuitive limit is the question of whether a given problem can be fruitfully reduced to a finite instance—but note that in the probabilistic setting we were able to deal with an infinite domain.

New Application Domains. Most of the results we have surveyed in this chapter focussed on voting rules. The method we presented is flexible enough to apply to other types of objects. Let us briefly sketch some ideas for future applications.

In Chapter 2 of this book, we have seen several axiomatic results in the theory of *multiwinner elections*; however impossibility theorems are notably missing. This could be a promising opportunity. To keep formula sizes tractable, it would be useful to consider only small committee sizes (such as $k = 2, 3$). It may also be fruitful to consider the approval setting, where voters submit dichotomous orders (see Chapter 2). In particular, the approval-based rules AV, MAV, and PAV all have different axiomatic strengths, and combining any two may lead to an interesting impossibility.

Matching theory is another plausible domain. In standard two-sided matching problems, there are known strategyproof mechanisms (such as one side of Gale–Shapley), and one could aim for impossibilities of a combination of strategyproofness with other axioms. Similarly, impossibility results could inform the theory of popular matchings (see Chapter 6). Finally, there is a large literature on the random assignment problem (see Chapter 1), which already contains several impossibilities. Computer-aided approaches could help strengthen and extend those results.

Judgement aggregation is about combining judgements about the truth-values of multiple logical formulas. The field has a rich axiomatic theory (see Endriss, 2016). Logic-based solvers may be particularly useful in this domain. A plausibly helpful tool would be the use of SMT solvers using a bitvector theory.

Argumentation theory is a further possible application domain. See, for example, the work by Booth et al. (2014), in which three-valued logics are applied to directed graphs, reminiscent of tournament solutions.

13.5 Conclusions

In the papers surveyed in this chapter, the application of computer-aided methods has lead to new insights for a range of questions in social choice theory that are of independent interest to the social choice community and unlikely to have been found without the help of computers.

Given the universality of the presented methods and their ease of adaptation (such as to "testing" of similar conjectures with minimal effort by replacing or altering some axioms), we anticipate these and similar techniques to yield further insights and to solve other open problems in social choice theory and related

research areas in the future. The breadth of results obtained so far supports this hypothesis.

Furthermore, we hope that the tutorial in this chapter will make the method accessible to many more researchers. It will be exciting to see it applied to new areas.

Acknowledgments

We thank Felix Brandt and Ulle Endriss for helpful comments.

Bibliography

G. Audemard and L. Simon. Predicting learnt clauses quality in modern SAT solvers. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 399–404, 2009.

G. Bachmeier, F. Brandt, C. Geist, P. Harrenstein, K. Kardel, D. Peters, and H. G. Seedig. k-majority digraphs and the hardness of voting with a constant number of voters. 2016. Working paper.

S. Barberà, W. Bossert, and P. K. Pattanaik. Ranking sets of objects. In S. Barberà, P. J. Hammond, and C. Seidl, editors, *Handbook of Utility Theory*, volume II, chapter 17, pages 893–977. Kluwer Academic Publishers, 2004.

A. Belov and J. Marques-Silva. MUSer2: An efficient MUS extractor. *Journal on Satisfiability, Boolean Modeling and Computation*, 8:123–128, 2012.

A. Biere. Lingeling, Plingeling and Treengeling entering the SAT competition 2013. In *Proceedings of the SAT Competition 2013*, pages 51–52, 2013.

R. Booth, E. Awad, and I. Rahwan. Interval methods for judgment aggregation in argumentation. In *Proceedings of the 5th International Workshop on Computational Social Choice (COMSOC)*, 2014.

F. Brandl, F. Brandt, C. Geist, and J. Hofbauer. Strategic abstention based on preference extensions: Positive results and computer-generated impossibilities. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 18–24. AAAI Press, 2015.

F. Brandl, F. Brandt, and C. Geist. Proving the incompatibility of efficiency and strategyproofness via SMT solving. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 116–122. AAAI Press, 2016.

F. Brandt and C. Geist. Finding strategyproof social choice functions via SAT solving. *Journal of Artificial Intelligence Research*, 55:565–602, 2016.

F. Brandt, C. Geist, and P. Harrenstein. A note on the McKelvey uncovered set and Pareto optimality. *Social Choice and Welfare*, 46(1):81–91, 2016.

F. Brandt, C. Geist, and D. Peters. Optimal bounds for the no-show paradox via SAT solving. *Mathematical Social Sciences*, 2017. Special Issue in Honor of Hervé Moulin. Forthcoming.

M. B. Caminati, M. Kerber, C. Lange, and C. Rowat. Sound auction specification and implementation. In *Proceedings of the 16th ACM Conference on Economics and Computation (ACM-EC)*, pages 547–564. ACM Press, 2015.

S. Chatterjee and A. Sen. Automated reasoning in social choice theory — some remarks. *Mathematics in Computer Science*, 8(1):5–10, 2014.

G. Ciná and U. Endriss. Proving classical theorems of social choice theory in modal logic. *Journal of Autonomous Agents and Multiagent Systems*, 30(5):963–989, 2016.

V. Conitzer and T. Sandholm. Complexity of mechanism design. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 103–110, 2002.

J. Drummond, A. Perrault, and F. Bacchus. 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)*, pages 518–525. AAAI Press, 2015.

M. Eberl. A formal proof of the incompatibility of SD-efficiency and SD-strategy-proofness. Bachelor's thesis, Technische Universität München, 2016.

U. Endriss. Judgment aggregation. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 17. Cambridge University Press, 2016.

P. C. Fishburn. *The Theory of Social Choice*. Princeton University Press, 1973.

A. Fréchette, N. Newman, and K. Leyton-Brown. Solving the station repacking problem. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2016.

P. Gärdenfors. On definitions of manipulation of social choice functions. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*. North-Holland, 1979.

C. Geist. Finding preference profiles of Condorcet dimension k via SAT. Technical report, http://arxiv.org/abs/1402.4303, 2014.

C. Geist and U. Endriss. Automated search for impossibility theorems in social choice theory: Ranking sets of objects. *Journal of Artificial Intelligence Research*, 40:143–174, 2011.

U. Grandi and U. Endriss. First-order logic formalisation of impossibility theorems in preference aggregation. *Journal of Philosophical Logic*, 42(4):595–618, 2013.

M. J. H. Heule, O. Kullmann, and V. W. Marek. Solving and verifying the Boolean Pythagorean triples problem via cube-and-conquer. In *Proceedings of the 19th International Conference on Theory and Applications of Satisfiability Testing*, volume 9710 of *Lecture Notes in Computer Science (LNCS)*, pages 228–245. Springer-Verlag, 2016.

M. Kerber, C. Lange, and C. Rowat. An introduction to mechanized reasoning. *Journal of Mathematical Economics*, 66:26–39, 2016.

B. Konev and A. Lisitsa. A SAT attack on the Erdős discrepancy conjecture. In *Theory and Applications of Satisfiability Testing–SAT 2014*, pages 219–226. Springer, 2014.

D. Kroening and O. Strichman. *Decision Procedures: An Algorithmic Point of View*. Texts in Theoretical Computer Science. Springer, 2016.

M. H. Liffiton, A. Previti, A. Malik, and J. Marques-Silva. Fast, flexible MUS enumeration. *Constraints*, pages 1–28, 2015.

T. Mennle and S. Seuken. The Pareto frontier for random mechanisms. Mimeo, 2015.

H. Moulin. Condorcet's principle implies the no show paradox. *Journal of Economic Theory*, 45:53–64, 1988.

T. Nipkow. Social choice theory in HOL: Arrow and Gibbard-Satterthwaite. *Journal of Automated Reasoning*, 43:289–304, 2009.

T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL – A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science (LNCS)*. Springer-Verlag, 2002.

D. Peters. Condorcet's principle and the preference reversal paradox. In *Proceedings of the 16th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*. 2017.

P. Tang. *Computer-aided Theorem Discovery - A New Adventure and its Application to Economic Theory*. PhD thesis, The Hong Kong University of Science and Technology (HKUST), 2010.

P. Tang and F. Lin. A computer-aided proof to Gibbard–Satterthwaite theorem. Technical report, 2008. URL `http://iiis.tsinghua.edu.cn/~kenshin/GS_proof.pdf`.

P. Tang and F. Lin. Computer-aided proofs of Arrow's and other impossibility theorems. *Artificial Intelligence*, 173(11):1041–1053, 2009.

P. Tang and F. Lin. Discovering theorems in game theory: Two-person games with unique pure nash equilibrium payoffs. *Artificial Intelligence*, 175(14–15): 2010–2020, 2011.

G. S. Tseitin. On the complexity of derivation in propositional calculus. In J. H. Siekmann and G. Wrightson, editors, *Automation of Reasoning*, pages 466–483. Springer, 1983.