Identifying k-Majority Digraphs via SAT Solving

Felix Brandt
Institut für Informatik
TU München
brandtf@in.tum.de

Christian Geist
Institut für Informatik
TU München
geist@in.tum.de

Hans Georg Seedig
Institut für Informatik
TU München
seedigh@in.tum.de

ABSTRACT

Many voting rules—including single-valued, set-valued, and probabilistic rules—*only* take into account the majority digraph. The contribution of this paper is twofold. First, we provide a surprisingly efficient implementation for computing the minimal number of voters that is required to induce a given digraph. This implementation relies on an encoding of the problem as a Boolean satisfiability (SAT) problem which is then solved by a SAT solver. Secondly, we experimentally evaluate how many voters are required to induce the majority digraphs of real-world and generated preference profiles. Our results are based on datasets from the PrefLib library and preferences generated using stochastic models such as impartial culture, impartial anonymous culture, Mallows mixtures, and spatial models. It turns out that all tournaments checked in these experiments can be induced by at most five voters whereas all other digraphs can be induced by at most eight voters. We also confirm a conjecture by Shepardson and Tovey by verifying that all tournaments with less than eight vertices can be induced by three voters.

1. INTRODUCTION

Perhaps one of the most natural ways to aggregate binary preferences from individual voters to a group of voters is *simple majority rule*, which prescribes that one alternative is socially preferred to another whenever a majority of voters prefers the former to the latter. Majority rule has an intuitive appeal to democratic principles, is easy to understand and—most importantly—satisfies some attractive formal properties [25]. Moreover, almost all common voting rules coincide with majority rule in the two-alternative case. It would therefore seem that the existence of a majority of individuals preferring alternative x to alternative y signifies something fundamental and generic about the group's preferences over x and y. Indeed, many voting rules—including single-valued, set-valued, and probabilistic rules—*only* take into account the majority digraph.

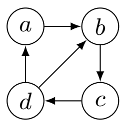
The central role of majority rule establishes an interesting connection between voting theory and graph theory. The earliest (and most fundamental) result in this context is *McGarvey's theorem*, which states that, given sufficiently many voters with linear preferences, *every* digraph may be induced by the majority rule [26]. In this paper, we will be concerned with the *minimal* number of voters $v(G)$ required to induce a given digraph G.

McGarvey's original construction requires two voters for each edge of the digraph, thus showing that $v(G) \leq 2\binom{n}{2}$ where n is the number of vertices of G. Consequently, this implied that the minimal number of voters $v(n)$ required to induce *any* digraph on n vertices is in $O(n^2)$. This bound was subsequently improved by Stearns [32], who showed that $v(n) = \Omega(n/\log n)$. Erdős and Moser [11] non-constructively provided a matching upper bound by proving that $v(n) = \Theta(n/\log n)$. More recently, Fiol [13] showed that $v(G) \leq n - \log n + 1$.

A digraph is a *k-majority digraph* if it can be induced by k voters. Interestingly, surprisingly little is known about the structure of k-majority digraphs. Dushnik and Miller [10] gave a complete characterization of 2-majority digraphs and Yannakakis [34] showed that the characterizing properties can be verified in polynomial time. Brandt et al. [8] provided a similar characterization for 3-majority digraphs. However, the computational complexity of checking whether a given digraph is a 3-majority digraph remains open. For the special case of *tournaments*, i.e., asymmetric and complete digraphs, Alon et al. [1] showed that the domination number of k-majority tournaments is bounded whereas Milans et al. [27] showed that every k-majority tournament contains a transitive subtournament whose size is at least polynomial in n.

The contribution of this paper is twofold. First, we provide a practical implementation for computing $v(G)$ for a given digraph G by encoding the problem as a Boolean satisfiability (SAT) problem which is then solved by a SAT solver. This technique turns out to be surprisingly efficient and easily outperforms an implementation for 3-majority digraphs based on the graph-theoretic characterization by Brandt et al. [8]. Secondly, we experimentally evaluate how many voters are required to induce the majority relations of real-world and generated preference profiles. Our results are based on datasets from the PrefLib library and preferences generated using stochastic models such as impartial culture, impartial anonymous culture, Mallows-ϕ, and spatial models. It turns out that all tournaments checked in these experiments are 5-majority tournaments whereas all other checked digraphs are 8-majority digraphs. Among other things, this shows that perhaps $v(G)$ itself may be used as a parameter to govern the generation of realistic preference profiles. We also confirm a conjecture by Shepardson and Tovey [31] by verifying that all tournaments with less than eight vertices are 3-majority digraphs.

Appears at: 1st Workshop on Exploring Beyond the Worst Case in Computational Social Choice. Held as part of the 13th International Conference on Autonomous Agents and Multiagent Systems. May 6th, 2014. Paris, France.



(a) digraph G

2	1	1	1	1
c	a	d	a	b
d	b	a	d	c
a	c	b	b	d
b	d	c	c	a

(b) preference profile R

Figure 1: A smallest 6-majority digraph with a minimal inducing preference profile.

2. PRELIMINARIES

Let A be a set of n alternatives and $K = \{1, \dots, k\}$ a set of voters. The preferences of voter $i \in K$ are represented by a linear (i.e., reflexive, complete, transitive, and antisymmetric) *preference relation* $R_i \subseteq A \times A$. The interpretation of $(a, b) \in R_i$, usually denoted by $a\ R_i\ b$, is that voter i values alternative a at least as much as alternative b. A *preference profile* $R = (R_1, \dots, R_k)$ is a k-tuple containing a preference relation R_i for each agent $i \in K$. For a preference profile R and two alternatives $a, b \in A$, the *majority margin* $g_R(a, b)$ is defined as the difference between the number of voters who prefer a to b and the number of voters who prefer b to a, i.e.,

$$g_R(a,b) = |\{i \in K \mid a\ R_i\ b\}| - |\{i \in K \mid b\ R_i\ a\}|.$$

Thus, $g_R(b, a) = -g_R(a, b)$ for all $a, b \in A$.

The *majority relation* $\succ_R$ of a given preference profile is defined as

$$a \succ_R b \quad \text{iff} \quad g_R(a,b) > 0.$$

Every majority relation $\succ_R$ is fully represented by a digraph G and we say that R *induces* G. If R has k voters, we say that G is *k-inducible*, or, equivalently, that G is a *k-majority digraph*.

If a digraph is complete, which is always the case if the number of voters is odd, we speak of a tournament $T = (A, \succ)$.

For any digraph G, by $v(G)$ we denote the minimal number of voters k such that G is a k-majority digraph. Occasionally, we will call this number the *voter complexity* of G.[1]

Example 1. *Consider the digraph G depicted on the left of Figure 1. We found that G is not 4-inducible. It cannot be 5-inducible either, because it is not a tournament as there is no strict relation between a and c. The profile R on the right of Figure 1, however, induces G and therefore G is a 6-majority digraph (or, equivalently, $v(G) = 6$). It turns out that G is a smallest digraph (in terms of the number of nodes) with voter complexity larger than 5.*

In this work, we address the computational problem of computing the voter complexity. To this end, we define the problem of checking whether for a given digraph G there exists a preference profile with k voters that induces G, i.e., whether G is a k-majority digraph.

[1]This complexity measure of digraphs can also be interpreted as a complexity measure for preference profiles. The voter complexity of a given preference profile is then simply defined as the voter complexity of the induced majority graph.

Name: Check-k-Majority
Instance: A digraph G and a positive integer k.
Question: Is G a k-majority digraph?

Note that the following two simple observations reduce the candidates for $v(G)$ to odd and even numbers, respectively, depending on whether G is a complete digraph or not.

Observation 1. *For all tournaments T, the voter complexity $v(T)$ is odd.*

Proof. Assume $v(T) = k$ was even. Then there exists a preference profile R with k voters that induces T. Since k is even, the majority margin must be even for every pair of alternatives and can furthermore never be 0 as T is a tournament. Therefore, removing any single voter from R gives a profile R' with just $k-1$ voters that still induces T, a contradiction. □

Observation 2. *For all incomplete digraphs G, the voter complexity $v(G)$ is even. It even holds that G is no k-majority digraph for k odd.*

Proof. This follows directly from the fact that for all preference profiles R with an odd number of voters k, the majority relation $\succ_R$ is complete and anti-symmetric (as no majority ties can occur). □

3. METHODOLOGY

The number of objects potentially involved in the Check-k-Majority problem are given in Table 1. It is immediately clear that a naïve algorithm will not solve the problem in a satisfactory manner. This section describes our algorithmic efforts to solve this problem for reasonably large instances.

3.1 Translation to propositional logic (SAT)

In order to answer Check-k-Majority, we follow a similar approach as Tang and Lin [33], Geist and Endriss [16], and Brandt and Geist [4]: we translate the problem to propositional logic (on a computer) and use state-of-the-art SAT solvers to find a solution. At a glance, the overall solving steps are shown in Algorithm 1.

Generally speaking, the problem at hand can be understood as the problem of finding a preference profile that satisfies certain conditions—here: inducing a given digraph. Thus, a satisfying instance of the propositional formula to be designed should represent a preference profile. To capture this, a surprisingly simple formalization involving just one type of variable suffices: in our encoding the boolean variable $r_{i,a,b}$ represents $a\ R_i\ b$, i.e., voter i ranking alternative a at least as high as alternative b. As it turns out, this one variable type also suffices for the additional condition of inducing the given digraph.

In more detail, the following three conditions/axioms need to be formalized:

1. All k voters have linear orders over the n alternatives as their preferences (short: linear preferences)

2. For each majority edge $x \succ y$ in the digraph, a majority of voters needs to prefer x over y (short: majority implications)

Preference profiles	$n=4$	$n=5$	$n=10$	$n=25$	$n=50$
$k=1$	24	120	$\sim 3.6 \cdot 10^{6}$	$\sim 1.6 \cdot 10^{25}$	$\sim 3.0 \cdot 10^{64}$
$k=3$	13,824	$\sim 1.7 \cdot 10^{6}$	$\sim 4.8 \cdot 10^{19}$	$\sim 3.7 \cdot 10^{75}$	$\sim 2.8 \cdot 10^{193}$
$k=5$	$\sim 8.0 \cdot 10^{6}$	$\sim 2.5 \cdot 10^{10}$	$\sim 6.3 \cdot 10^{32}$	$\sim 9.0 \cdot 10^{125}$	$\sim 2.6 \cdot 10^{322}$
Tournaments (unlabeled)	4	12	$\sim 9.7 \cdot 10^{6}$	$\sim 1.3 \cdot 10^{65}$	$\sim 1.9 \cdot 10^{305}$

Table 1: Number of objects involved in the CHECK-k-MAJORITY problem for one, three, and five voters.

```
Input: digraph (A, ≻), positive integer k
Output: whether (A, ≻) is a k-majority digraph
/* Encoding of problem in CNF                         */
File cnfFile;
foreach voter i do
    cnfFile += Encoder.reflexivePreferences(i);
    cnfFile += Encoder.completePreferences(i);
    cnfFile += Encoder.transitivePreferences(i);
    cnfFile += Encoder.antisymmetricPreferences(i);
cnfFile += Encoder.majorityImplications((A, ≻));
if ≻ is not complete then
    cnfFile +=
    Encoder.indifferenceImplications((A, ≻));
/* SAT solving                                        */
satisfiable = SATsolver.solve(cnfFile);
if instance is satisfiable then
    return true;
else
    return false
```

Algorithm 1: SAT-CHECK-k-MAJORITY

3. For each missing edge ($x \not\succ y$ and $y \not\succ x$) in the digraph, *exactly* half the voters need to prefer x over y (short: indifference implications)[2]

For the first axiom, we encode reflexivity, completeness, transitivity, and anti-symmetry of the relation R_i for all voters i. The complete translation to CNF (conjunctive normal form, the established standard input format for SAT solvers) is given exemplarily for the case of transitivity; the other axioms are converted analogously.

In formal terms transitivity can be written as

$$
\begin{aligned}
& (\forall i)(\forall x,y,z)\,(x\ R_i\ y \wedge y\ R_i\ z \rightarrow x\ R_i\ z) \\
\equiv\ & (\forall i)(\forall x,y,z)\,(r_{i,x,y} \wedge r_{i,y,z} \rightarrow r_{i,x,z}) \\
\equiv\ & \bigwedge_i \bigwedge_{x,y,z} (\neg\,(r_{i,x,y} \wedge r_{i,y,z}) \vee r_{i,x,z}) \\
\equiv\ & \bigwedge_i \bigwedge_{x,y,z} (\neg r_{i,x,y} \vee \neg r_{i,y,z} \vee r_{i,x,z})\,,
\end{aligned}
$$

which then translates to the pseudo code in Algorithm 2 for generating the CNF file. The key in the translation of the inherently higher order axioms to propositional logic is (as pointed out by Geist and Endriss [16] already) that because of finite domains, all quantifiers can be replaced by finite conjunctions or disjunctions, respectively.

In all algorithms, a subroutine $\mathrm{r}(i,x,y)$ takes care of the compact enumeration of variables.[3]

```
foreach Voter i do
    foreach Alternative x do
        foreach Alternative y do
            foreach Alternative z do
                variable_not(r(i, x, y));
                variable_not(r(i, y, z));
                variable(r(i, x, z));
                newClause;
```

Algorithm 2: Encoding of transitivity of individual preferences

The axioms of majority and indifference implications can be formalized in a similar fashion. We describe the translation for the majority implications here; the procedure for the indifference implications (needed for incomplete digraphs) is analogous again. In the following, we denote the smallest number of voters required for a positive majority margin by $m(k) := \lfloor k \cdot \frac{1}{2} \rfloor + 1$. Note that then, because of anti-symmetry of the individual preferences, for $x \succ y$ it suffices that there exists a set of $m(k)$ many voters who prefer x to y. In formal terms:

$$
\begin{aligned}
& (\forall x,y)\,(x \succ y \rightarrow |\{i \mid x\ R_i\ y\}| > |\{i \mid y\ R_i\ x\}|) \\
\equiv\ & (\forall x,y)\,(x \succ y \rightarrow |\{i \mid x\ R_i\ y\}| \geq s(n)) \\
\equiv\ & (\forall x,y)\,(x \succ y \rightarrow \\
& \qquad (\exists M \subseteq K)|M| = m(k) \wedge (\forall i \in M) x\ R_i\ y) \\
\equiv\ & \bigwedge_{x \succ y} \bigvee_{|M|=m(k)} \bigwedge_{i \in M} r_{i,x,y}.
\end{aligned}
$$

In order to avoid an exponential blow-up when converting this formula to CNF, variable replacement (a standard procedure also known as Tseitin transformation) is applied. In our case, we replaced $\bigwedge_{i \in M} r_{i,x,y}$ by new variables of the form $h_{M,x,y}$ and introduced the following defining clauses:

$$
\begin{aligned}
& \bigwedge_M \bigwedge_{x,y} \left(h_{M,x,y} \rightarrow \bigwedge_{i \in M} r_{i,x,y} \right) \\
\equiv\ & \bigwedge_M \bigwedge_{x,y} \left(\neg h_{M,x,y} \vee \bigwedge_{i \in M} r_{i,x,y} \right) \\
\equiv\ & \bigwedge_M \bigwedge_{x,y} \bigwedge_{i \in M} (\neg h_{M,x,y} \vee r_{i,x,y})\,.
\end{aligned}
$$

In this case, the helper variables even have an intuitive meaning as $h_{M,x,y}$ enforces that all the voters $i \in M$ prefer alternative x over alternative y.

[2] Note that this axiom is only required for incomplete digraphs.

[3] The DIMACS CNF format only allows for integer names of variables. But since we know in advance how many voters and alternatives there are, we can simply use a standard enumeration method for tuples of objects.

Note that the conditions like $|M| = m(k)$ can easily be fulfilled during generation of the corresponding CNF formula on a computer. For enumerating all subsets of voters of a given size we, for instance, used Gosper's Hack [18]. The corresponding pseudo code for majority implications can be found in Algorithm 3.

```
foreach Pair of alternatives x ≻ y do
    foreach M ⊆ K, |M| = m(k) do
        variable(h(M, x, y));
    newClause;
/* start of helper variable definition          */
foreach Pair of alternatives x ≻ y do
    foreach M ⊆ K, |M| = m(k) do
        foreach i ∈ M do
            variable_not(h(M, x, y));
            variable(r(i, x, y));
            newClause;
```

Algorithm 3: Encoding of majority implications

This encoding leads to a total of $k \cdot n^2 + \binom{k}{m(k)} \cdot n^2 = n^2 \cdot \left(k + \binom{k}{m(k)}\right)$ variables for the case of tournaments and $n^2 \cdot \left(k + \binom{k}{m(k)} + \binom{k}{k/2}\right)$ variables for incomplete digraphs.The number of clauses is equal to $k \cdot (n^3 + n^2) + \frac{n^2-n}{2} \cdot \left(1 + \binom{k}{m(k)} \cdot m(k)\right)$ and at most $k \cdot (n^3 + n^2) + (n^2 - n) \cdot \left(1 + \binom{k}{k/2} \cdot \frac{k}{2}\right)$ for tournaments and incomplete digraphs, respectively.

With all axioms formalized in propositional logic, we are now ready to analyze arbitrary digraphs G for their voter complexity $v(G)$. Before we do so, however, we describe an optimization technique for tournament graphs, which, for certain instances, speeds up the computation significantly.

3.2 Optimized computation for tournaments via components

An important structural property in the context of tournaments is whether a tournament admits a non-trivial decomposition. Brandt et al. [7] show that this decomposition allows for a recursive computation of certain concepts, which is particularly helpful if the original computation is costly for large instances.[4] We are now going to prove that a similar optimization can be carried out for the computation of the voter complexity $v(T)$ of a given tournament T. In particular, we show that the voter complexity of a tournament is equal to the maximum of the voter complexities of its components and the corresponding summary.

In formal terms, a non-empty subset B of A is a *component* of a tournament $T = (A, \succ)$ if for all $a \in A \setminus B$ either $B \succ a$ or $a \succ B$, where $B \succ a$ stands for $(\forall b \in B) b \succ a$. A *decomposition* of T is a set of pairwise disjoint components $\{B_1, \ldots, B_p\}$ of T such that $A = \bigcup_{j=1}^{p} B_j$. The decomposition is *proper* if $p > 1$ and not all B_j are singletons. Every tournament admits a decomposition that is minimal in a well-defined sense [20]. Given a particular decomposition $\tilde{B} = \{B_1, \ldots, B_p\}$, the *summary* of T with respect to $\tilde{B}$ is defined as the tournament $T_B = (\{1, \ldots, p\}, \tilde{\succ})$ on the individual components rather than the alternatives, i.e.,

$$q \mathrel{\tilde{\succ}} r \quad \text{if and only if} \quad B_q \succ B_r.$$

Each component B_j (including A) naturally induces a subtournament T_{B_j} which is the summary of $T|_{B_j}$ with respect to its minimal decomposition.

The following lemma then enables the recursive computation of $v(T)$ along the component structure of T:

LEMMA 1. *Let T be a tournament and $\tilde{B} = \{B_1, \ldots, B_p\}$ a decomposition of T. Then*

$$v(T) = \max_j \{v(T_{B_j}), v(T_B)\}.$$

PROOF. Let R be a minimal profile inducing T. Then, $R|_{B_j}$ induces T_{B_j} for every B_j establishing $v(T) \geq v(T_{B_j})$. That $v(T) \geq v(T_B)$ holds is also easy to see by considering a variant of R in which from each component all but one node are arbitrarily chosen and removed. The remaining profile then induces T_B. For the other direction, let $v'(T) = \max_j \{v(T_{B_j}), v(T_B)\}$. We know, by Observation 1, that $v(T')$ and every $v(T_{B_j})$ is odd, as these are all tournaments. Each T_{B_j} (and T_B) has a minimal profile R^j (and R, respectively). We can add pairs of voters with opposing preferences to each profile without changing its majority relation. This way, we get profiles R'^j (and R') that still induce T_{B_j} (or T_B) but now all have the same number of voters $v'(T)$. Now, create a new profile $\hat{R}$ from R' in which R_i^j replaced alternative j as a segment in R'_i for each voter i and every alternative j as in [19]. It is easy to check that $\hat{R}$ has $v'(T)$ voters and still induces T, i.e., $v(T) \geq v'(T) = v(T_{B_j})$. □

We have implemented this optimization and found that many real-world majority digraphs exhibit proper decompositions, speeding up the computation of SAT-CHECK-k-MAJORITY.

3.3 Data sources and method of analysis

In the preference library PREFLIB [23], scholars have contributed data sets from real world scenarios ranging from preferences over movies or sushi via Formula 1 championship results to real election data. Accordingly, the number of voters whose preferences originally induced these data sets vary heavily between 4 and 44000. At the time of writing, PREFLIB contained 354 tournaments induced from pairwise majority comparisons as well as 185 incomplete majority digraphs.

Additionally, we consider *stochastic* models to generate tournaments of a given size n. Many different models for linear preferences (or orderings) have been considered in the literature. We refer the interested reader to [9, 22, 24, 6]. In this work, we decided to examine tournaments generated with five different stochastic models.

In the *uniform random tournament model*, the same probability is assigned to each *labeled* tournament of size n, i.e.,

$$\Pr(T) = \frac{1}{2^{\binom{n}{2}}} \text{ for each } T \text{ with } |T| = n.$$

In all of the remaining models, we sample preference profiles and work with the tournament induced by the majority relation. In accordance with [6], we generated profiles with 51 voters.

[4] As Brandt et al. [7] point out, the decomposition of a tournament can be computed in linear time.

The *impartial culture model* (IC) is the most widely-studied model for individual preferences in social choice. It assumes that every possible preference ordering has the same probability of $\frac{1}{n!}$. If we add anonymity by having indistinguishable voters, the set of profiles is partitioned into equivalence classes. In the *impartial anonymous culture* (IAC), each of these equivalence classes is chosen with equal probability.

In *Mallows-ϕ model* [21], the distance to a reference ranking is measured by means of the Kendall-tau distance which counts the number of pairwise disagreements. Let R_0 be the reference ranking. Then, the Kendall-tau distance of a preference ranking R to R_0 is

$$\tau(R, R_0) = \binom{n}{2} - |R \cap R_0|.$$

According to the model, this induces the probability of a voter having R as his preferences to be

$$\Pr(R) = \frac{\phi^{\tau(R,R_0)}}{C}$$

where C is a normalization constant and $\phi \in (0, 1]$ is a dispersion parameter. Small values for ϕ put most of the probability on rankings very close to R_0 whereas for $\phi = 1$ the model coincides with IC.

A very different kind of model is the *spatial model*. Here, alternatives and voters are uniformly at random placed in a multi-dimensional space and the voters' preferences are determined by the (Euclidian) distanced to the alternatives. The spatial model has played an important role in political and social choice theory where the dimensions are interpreted as different aspects or properties of the alternatives (see, e.g., [28, 2]).

4. RESULTS

All experiments were run on a Intel Core i5, 2.66GHz (quad-core) machine with 12 GB RAM using the SAT solver PLINGELING [3].

4.1 Exhaustive analysis

We generated all tournaments with up to 10 alternatives and found that all of these are 5-inducible. In fact, all tournaments of size up to seven are even 3-inducible, confirming a conjecture by Shepardson and Tovey [31]. They also showed that there exist tournaments of size 8 that are not 3-inducible. We find that the exact number of such tournaments is 96 (out of 6880).

Brandt and Seedig [5] presented a highly structured tournament on 24 alternatives that serves as the current minimal counterexample to a now disproved conjecture by Schwartz [30] in social choice theory. We found it to be a 5-majority tournament, implying that the negative theoretical consequences of the counterexample already hold for scenarios with only 5 voters (and at least 24 alternatives).

4.2 Empirical analysis

Among the tournaments in PrefLib, 58 are 3-inducible. Out of the two largest tournaments in the data set with 240 and 242 alternatives, respectively, the first is a 5-majority tournament while on the second the SAT solver did not terminate within one day. The remaining tournaments are transitive and thus 1-inducible. Therefore, all checkable tournaments in PrefLib are inducible by only 5 voters.

For the non-complete majority digraphs in PrefLib, we found that the indifference constraints which are imposed on missing edges change the picture. Not only does it negatively affect the running time of SAT-Check-k-Majority in comparison to tournaments which made us restrict our attention to instances with at most 40 alternatives, but it also seems to result in higher voter complexities of up to 8 among the 85 feasible instances. However, given that the number of voters in the profiles that originally induced these majority digraphs are often in the hundreds or thousands, we still consider these low voter complexities.

n	uniform	IC	IAC	Mallows-ϕ ($\phi = 0.95$)	spatial (dim = 2)
3	1.40	1.13	1.13	1.13	1.00
5	3.00	1.67	2.13	1.33	1.13
7	3.00	2.67	2.67	2.47	1.33
9	3.13	3.00	3.00	2.67	1.60
11	3.93	3.07	3.00	2.87	2.33
13	4.80	3.07	3.20	2.93	2.53
15	5.00	3.27	3.40	3.00	2.67
17	5.00	3.40	3.80	2.93	2.80
19	5.00	4.27	4.20	3.00	2.80
21	5.00	4.47	4.33	3.00	2.87

Table 2: Average voter complexity in tournaments generated by stochastic (preference) models. The given values are averaged over 30 samples each.

4.3 Stochastic analysis

For up to 21 alternatives, we sampled preference profiles (each consisting of 51 voters[5]) from the aforementioned stochastic models and examined the corresponding majority graphs for their voter complexity using SAT-Check-k-Majority. The average complexities over 30 instances of each size are shown in Table 2. We see that the unbiased models (IC, IAC, uniform) tend to induce digraphs with higher voter complexity. We encountered no tournament that was not a 5-majority tournament.[6]

4.4 Runtime analysis

A characterization by Brandt et al. [8] of 3-majority digraphs allows for a straightforward algorithm, which is expected to have a much better running time than any naïve implementation enumerating all preference profiles (also compare Table 1). The characterization is given in Lemma 2 below, as is the corresponding algorithm 2-Partition-Check-3-Majority (Algorithm 4). Besides enumerating all 2-partitions of the majority relations, the only non-trivial part is to check whether a relation has a transitive reorientation. This can be done efficiently using an algorithm by Pnueli et al. [29].

We compared the running times of 2-Partition-Check-3-Majority with the ones of our implementation via SAT

[5]In another study [6], this size turned out to be sufficiently large to discriminate the different underlying stochastic models.

[6]Our efforts also included checking more than 8 million uniform random tournaments with 12 alternatives.

as described in Section 3.1 (see also Algorithm 1).[7]

Surprisingly, it turns out that—even though it is much more universal—SAT-Check-3-Majority offers significantly better running times. Preliminary data is displayed in Table 3. Note that in addition to being more efficient, SAT-Check-k-Majority is even able to return a preference profile with k voters that induces the given digraph (without the need for additional computations).

Further runtimes, which exhibit the practical power of our SAT approach (and its limits), can be obtained from Table 4.

Lemma 2 (Brandt et al.). *A digraph $(A, \succ)$ is a 3-majority digraph if and only if $\succ$ is complete and there are disjoint sets $\succ_1, \succ_2$ with $\succ = (\succ_1 \cup \succ_2)$ such that*

- *$(A, \succ_1)$ is a 2-majority digraph and*
- *$\succ_2$ is acyclic.*

Whether $(A, \succ_1)$ is a 2-majority digraph can efficiently be checked [34] via the following characterization by Dushnik and Miller [10]:

Lemma 3 (Dushnik and Miller). *A digraph $(A, \succ)$ is a 2-majority digraph if and only if*

- *$\succ$ is transitive and*
- *there exists a reorientation of $\big((A \times A) \setminus (\succ \cup \succ^{-1})\big)$ that is transitive and asymmetric.*

```
Input: digraph (A, ≻)
Output: whether (A, ≻) is a 3-majority digraph
if ≻ is complete then
    foreach 2-partition {≻1, ≻2} of ≻ do
        if ≻1 is transitive and ≻2 is acyclic and ≻2 has
        a transitive reorientation then
            return true;
        else
            return false;
else
    return false;
```

Algorithm 4: 2-Partition-Check-3-Majority

5. OUTLOOK AND FUTURE WORK

The following two insights of this work have been most surprising to us.

- First, our SAT-based implementation significantly outperforms the best direct algorithm known to us, while at the same time being much more flexible and powerful.[8]
- Second, the voter complexity of any majority digraph we could analyze does not exceed five for tournaments, and eight for incomplete digraphs, respectively.

[7]As a programming language Java was used in both cases.

[8]In the sense that it can also solve instances for $k \geq 3$.

Both of these points offer many directions for future work. Our implementation might be useful to find concrete tournaments that are not k-inducible, a problem that has occupied graph theorists. For example, the order of the smallest tournament that is not 5-inducible is currently unknown. Analytical results by Alon et al. [1], Graham and Spencer [17], and Fidler [12] can be used to narrow down the search for such tournaments. Preliminary results suggest that quadratic residue tournaments are good candidates for tournaments that can only be induced by a large number of voters. We intend to further pursue this direction in future work.

As other solving techniques are concerned, a natural choice for the problem at hand are techniques that can handle cardinality constraints natively (rather than encoding them in SAT/CNF as we did). ASP (answer set programming, see, e.g., Gebser et al. [15]) is an example of such a technique. We were able to obtain preliminary results using an ASP formulation of the problem (see Figure 2) and a corresponding solver (clasp with grounder Gringo [14]). While due to its richer problem description language (which also includes cardinality constraints) the formalization is much more compact than the corresponding SAT/CNF formulation, interestingly, performance appears to be similar or even slightly worse compared to current SAT solvers. Other solvers with cardinality constraints, however, might lead to different performance results.

Our approach can also be used to treat a range of related problems and questions. For instance, one could define natural variants of the notion of k-majority digraphs such as voters having weak (i.e., ties are allowed) or even incomplete preferences. Because of the high flexibility of our SAT formalization, one can easily apply the same method to analyze these related concepts and questions.[9] Even weighted majority graphs, i.e., graphs which carry the majority margin as weights on edges, can be analyzed regarding their voter complexity by slightly adapting our SAT or ASP encodings.

REFERENCES

[1] N. Alon, G. Brightwell, H. A. Kierstead, A. V. Kostochka, and P. Winkler. Dominating setse in k-majority tournaments. *Journal of Combinatorial Theory Series B*, 96:374–387, 2006.

[2] D. Austen-Smith and J. S. Banks. *Positive Political Theory I: Collective Preference.* University of Michigan Press, 2000.

[3] A. Biere. Lingeling, plingeling and treengeling entering the sat competition 2013. In A. Balint, A. Belov, M. Heule, and M. Järvisalo, editors, *Proceedings of the SAT Competition 2013*, pages 51–52, 2013.

[4] F. Brandt and C. Geist. Finding strategyproof social choice functions via SAT solving. In *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS).* IFAAMAS, 2014. Forthcoming.

[5] F. Brandt and H. G. Seedig. A tournament of order 24 with two disjoint TEQ-retentive sets. Technical report, http://arxiv.org/abs/1302.5592, 2013.

[9]For the two suggested variants, deleting axioms from the formalization suffices.

algorithm	5	6	7	8	9	10	20	50	100
SAT	< 0.1s	< 0.1s	< 0.1s	< 0.1s	< 0.1s	< 0.1s	0.1s	1.5s	12.5s
2-Partition	< 0.1s	0.1s	2s	1200s	—	—	—	—	—

Table 3: Runtime comparison of the SAT implementation for $k = 3$ and 2-Partition-Check-3-Majority for complete digraphs (tournaments) of different sizes n with a cutoff time of one hour.

$n \backslash k$	3	4	5	6	7	8	9	10	11	12
3	.04	.04	.03	.04	.04	.04	.04	.05	.08	.10
4	.03	.04	.03	.04	.04	.04	.05	.07	.10	.18
5	.03	.04	.03	.04	.06	.05	.06	.09	.16	.35
6	.03	.04	.04	.04	.05	.06	.08	.12	.27	.63
7	.04	.04	.04	.05	.05	.07	.10	.17	.45	1.10
8	.04	.05	.05	.05	.07	.08	.13	.23	.69	1.80
9	.04	.05	.05	.64	.07	.10	.17	.33	1.06	2.83
10	.05	.05	.06	.67	.09	.12	.23	.46	1.56	4.25
11	.06	.06	.06	1.92	.10	.14	.30	.63	2.22	6.37
12	.06	.07	.07	3.35	.12	.19	.40	.85	3.18	8.48
13	.07	.07	.09	3.93	.15	.27	.52	1.16	4.44	12.30
14	.07	.09	.10	4.15	.18	.36	.64	1.51	5.99	16.84
15	.08	.10	.13	3.89	.21	.88	.79	2.22	7.67	—
16	.09	.11	.14	4.12	.25	4.55	.99	2.90	9.80	—
17	.10	.12	.19	4.41	.29	7.15	1.23	4.69	12.48	—
18	.11	.14	.23	4.76	.35	17.51	1.53	8.25	15.97	—
19	.12	.15	.35	4.97	.43	—	1.80	—	19.99	—
20	.13	.17	.54	5.04	.47	—	2.21	—	—	—
21	.14	.18	5.87	6.15	.63	—	2.71	—	—	—
22	.16	.20	11.07	5.43	.96	—	3.24	—	—	—
23	.17	.23	18.95	5.76	1.57	—	4.12	—	—	—
24	.20	.26	—	5.87	2.56	—	4.60	—	—	—
25	.22	.29	—	6.12	4.21	—	5.85	—	—	—

Table 4: Runtime in seconds of SAT-Check-k-Majority for different number of alternatives and different number of voters k when average runtimes did not exceed 20 seconds. For this table, averages were taken over 5 samples from the uniform random tournament model.

[6] F. Brandt and H. G. Seedig. On the discriminative power of tournament solutions. In *Proceedings of the 1st AAMAS Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE)*, 2014. Forthcoming.

[7] F. Brandt, M. Brill, and H. G. Seedig. On the fixed-parameter tractability of composition-consistent tournament solutions. In T. Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 85–90. AAAI Press, 2011.

[8] F. Brandt, P. Harrenstein, K. Kardel, and H. G. Seedig. It only takes a few: On the hardness of voting with a constant number of agents. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 375–382. IFAAMAS, 2013.

[9] D. E. Critchlow, M. A. Fligner, and J. S. Verducci. Probability models on rankings. *Journal of Mathematical Psychology*, 35:294–318, 1991.

[10] B. Dushnik and E. W. Miller. Partially ordered sets. *American Journal of Mathematics*, 63(3):600–610, 1941.

[11] P. Erdős and L. Moser. On the representation of directed graphs as unions of orderings. *Publ. Math. Inst. Hungar. Acad. Sci.*, 9:125–132, 1964.

[12] D. Fidler. A recurrence for bounds on dominating sets in k-majority tournaments. *The Electronic Journal of Combinatorics*, 18(1), 2011.

[13] M. A. Fiol. A note on the voting problem. *Stochastica*, XIII-1:155–158, 1992.

[14] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and M. Schneider. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24 (2):107–124, 2011.

[15] M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub. Answer set solving in practice. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(3):1–238, 2012.

[16] 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.

[17] R. L. Graham and J. H. Spencer. A constructive solution to a tournament problem. *Canadian Mathematical Bulletin*, 14(1):45–48, 1971.

```
%Alternatives
#const m=19.
alt(1..m).

%Number of voters
#const n=6.
voter(1..n).

#const simple_majority=4.

#const indifference_majority=3.

%Completeness and Antisymmetry
1{r(I,X,Y);r(I,Y,X)}1 :- voter(I), alt(X;Y), X!=Y.

%Reflexivity
r(I,X,X) :- voter(I), alt(X).

%Transitivity
r(I,X,Z) :- r(I,X,Y), r(I,Y,Z).

%Majority implications
simple_majority{r(I,X,Y):voter(I)} :- g(X,Y), X!=Y.

%Indifference implications
indifference_majority{r(I,X,Y):voter(I)}
indifference_majority :- i(X,Y), X!=Y.

g(1,2).
g(1,3).
g(1,4).
# (...) graph encoding mostly ommitted
g(19,14).
g(19,18).


#show r/3.
```

Figure 2: Problem description in ASP for $k = 6$ and a majority digraph with $n = 19$ nodes. Parts of the majority graph have been omitted to increase readability.

[18] D. E. Knuth. *Combinatorial Algorithms*, volume 4A, part 1 of *The Art of Computer Programming*. Addison-Wesley, 2011.

[19] G. Laffond, J. Lainé, and J.-F. Laslier. Composition-consistent tournament solutions and social choice functions. *Social Choice and Welfare*, 13:75–93, 1996.

[20] J.-F. Laslier. *Tournament Solutions and Majority Voting*. Springer-Verlag, 1997.

[21] C. L. Mallows. Non-null ranking models. *Biometrika*, 44(1/2):114–130, 1957.

[22] J. I. Marden. *Analyzing and Modeling Rank Data*. Number 64 in Monographs on Statistics and Applied Probability. Chapman & Hall, 1995.

[23] N. Mattei and T. Walsh. PrefLib: A library for preference data http://www.preflib.org. In P. Perny, M. Pirlot, and A. Tsoukiàs, editors, *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT)*, volume 8176 of *Lecture Notes in Computer Science (LNCS)*, pages 259–270. Springer, 2013.

[24] N. Mattei, J. Forshee, and J. Goldsmith. An empirical study of voting rules and manipulation with large datasets. In *Proceedings of the 4th International Workshop on Computational Social Choice (COMSOC)*, 2012.

[25] K. May. A set of independent, necessary and sufficient conditions for simple majority decisions. *Econometrica*, 20:680–684, 1952.

[26] D. C. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.

[27] K. G. Milans, D. H. Schreiber, and D. B. West. Acyclic sets in k-majority tournaments. *The Electronic Journal of Combinatorics*, 18(1), 2011.

[28] P. Ordeshook. The spatial analysis of elections and committees: four decades of research. Mimeo, 1993.

[29] A. Pnueli, A. Lempel, and S. Even. Transitive orientation of graphs and identification of permutation graphs. *Canadian Journal of Mathematics*, 23:160–175, 1971.

[30] T. Schwartz. Cyclic tournaments and cooperative majority voting: A solution. *Social Choice and Welfare*, 7 (1):19–29, 1990.

[31] D. Shepardson and C. A. Tovey. Smallest tournament not realizable by $\frac{2}{3}$-majority voting. *Social Choice and Welfare*, 33(3):495–503, 2009.

[32] R. Stearns. The voting problem. *American Mathematical Monthly*, 66(9):761–763, 1959.

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

[34] M. Yannakakis. The complexity of the partial order dimension problem. *SIAM Journal on Algebraic and Discrete Methods*, 3(3):351–358, 1982.