Parameterized Complexity Results for Agenda Safety in Judgment Aggregation

Ulle Endriss
ILLC, University of Amsterdam
ulle.endriss@uva.nl

Ronald de Haan
Technische Universität Wien
dehaan@kr.tuwien.ac.at

Stefan Szeider
Technische Universität Wien
stefan@szeider.net

ABSTRACT

Many problems arising in computational social choice are of high computational complexity, and some are located at higher levels of the Polynomial Hierarchy. We argue that a parameterized complexity analysis provides valuable insight into the factors contributing to the complexity of these problems, and can lead to practically useful algorithms. As a case study, we consider the problem of agenda safety for the majority rule in judgment aggregation, consider several natural parameters for this problem, and determine the parameterized complexity for each of these. Our analysis is aimed at obtaining fixed-parameter tractable (fpt) algorithms that use a small number of calls to a SAT solver. We identify several positive results, including several results where the problem can be fpt-reduced to a single SAT instance. In addition, we identify several negative results. We hope that this work may help initiate a structured parameterized complexity investigation of problems arising in the field of computational social choice that are located at higher levels of the Polynomial Hierarchy.

Categories and Subject Descriptors

F.2 [**Analysis of Algorithms and Problem Complexity**]: General

General Terms

Theory

Keywords

Judgment Aggregation; Agenda Safety; Complexity Theory; Parameterized Complexity; Treewidth

1. INTRODUCTION

The field of computational social choice studies the interface of social choice theory and computer science. In particular, it is concerned with investigating properties of computational tasks related to procedures for collective decision making. Some of these computational tasks have a computational complexity that is 'beyond NP', and are thus considered to be highly intractable (cf. [2, 11, 29, 30]). We argue that the complexity analysis of problems arising in computational social choice that are 'beyond NP' benefits from a parameterized complexity approach [18, 19, 22, 37]. Recent advances in parameterized complexity theory [28] enable an investigation of the restrictions that allow an encoding of problems 'beyond NP' into the Boolean satisfiability problem (SAT). With the success that modern SAT solving algorithms have had in many practical settings over the last two decades [34, 39], this might lead to practically useful algorithms for problems that are traditionally considered to be highly intractable.

As a case study to underpin our argument, we consider the computational complexity of the problem of *agenda safety*, which is a computational problem that arises in the domain of judgment aggregation. Judgment aggregation studies the properties of procedures that combine the individual judgments on a set of related propositions (the *agenda*) of the members of a group into a collective judgment reflecting the views of the group as a whole [33]. Such procedures might, in general, yield inconsistent combined judgments. An example of a procedure whose outcome can be inconsistent is the *majority rule*, where for each issue the collective judgment agrees with the majority of the individual judgments on this issue. Because of such possible inconsistencies, it is useful to determine for a given agenda and some aggregation procedure whether there exists no combination of individual judgments such that the outcome of the procedure is inconsistent (we say that the agenda is *safe* if this is the case). This is relevant, for instance, in the setting of multiagent systems where agents need to coordinate their beliefs, intentions and actions repeatedly [42]. In this scenario, we might have to check whether the logical structure of the issues to be decided upon collectively is such that the majority decision can be guaranteed to be consistent for any combination of consistent choices made by the individual agents. The problem of agenda safety for the majority rule is complete for the second level of the Polynomial Hierarchy (PH) [20], and is thus 'beyond NP.'

Instances of hard computational problems that occur in practice often exhibit some kind of structure. A classical complexity analysis is insensitive to any such structure. A parameterized complexity analysis, on the other hand, can take into account different forms of structure in the problem instances, by means of problem parameters. The idea underlying parameterized complexity theory is that such parameters are expected to be small in problem instances occurring in practice. By restricting the high complexity of a problem to the parameter only, these structured instances

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

of hard computational problems can often be solved reasonably efficiently. There has been a lot of research in the field of parameterized complexity over the last two decades [10]. Most of this research is aimed at problems that are in NP. Recently, tools have been developed to analyze the parameterized complexity of problems that are located higher in the PH [26, 27, 28]. The paradigm of parameterized complexity has been used to examine many problems in computational social choice [3, 4, 5, 17].

Contributions.

Concretely, we investigate what kind of structure helps to decrease the computational complexity of the problem of agenda safety for the majority rule. We do this by studying several natural parameterizations of the problem. The main concept of tractability that we have in mind is based on algorithms that run efficiently for small parameter values, and that use only a small number of SAT calls (depending on the parameter value only), preferably only a single call. Such parameterized algorithms are an improvement over polynomial-time algorithms, because the problem cannot be solved by a polynomial-time algorithm that can make calls to a SAT solver (unless the PH collapses). This notion of tractability is motivated by the enormous practical success of modern SAT solvers [8, 24, 34, 39]. For precise definitions, we refer to Section 2.

Several parameterizations that we consider correspond to simple syntactic restrictions on the agenda (i.e., bounds on the size of formulas, bounds on variable occurrence, and bounds on the number of formulas). Another several parameterizations that we consider capture structure in the agenda in terms of the 'tree-likeness' of various graphs associated to the agenda. Yet another parameterization corresponds to a bound on the size of counterexamples (to the logical characterization of agenda safety). These parameterizations have been applied successfully in other domains [6, 7, 25]. An overview of complexity results for these parameterizations can be found in Table 1.

We identify several positive cases, where structure present in the problem input allows us to solve the problem using a small number of SAT calls, in several cases even using a single call. These positive results could lead to algorithms that perform well in practice. Additionally, our parameterized complexity analysis allows us to pinpoint exactly what aspects of the problem play what role in the high computational complexity of the problem, and it helps to determine what algorithmic approach is well suited to solve the problem in practical settings. We hope that this work can help initiate a structured parameterized complexity investigation of problems arising in the field of computational social choice that are located at higher levels of the PH.

2. PRELIMINARIES

In this section, we formally define the problem of agenda safety and we provide a logical characterization of the problem for a particular aggregation procedure. Moreover, we review notions from complexity theory.

Propositional Logic and Agenda Safety.

A *literal* is a propositional variable x or a negated variable $\neg x$. A *clause* is a finite set of literals, not containing a complementary pair x, $\neg x$, and is interpreted as the disjunction of these literals. A formula in *conjunctive normal form (CNF)* is a finite set of clauses, interpreted as the conjunction of these clauses. We define the *size* $||\varphi||$ of a CNF formula φ to be $\sum_{c \in \varphi} |c|$; the number of clauses of φ is denoted by $|\varphi|$. For a CNF formula φ, the set $\text{Var}(\varphi)$ denotes the set of all variables x such that some clause of φ contains x or $\neg x$. We say that a clause is a *Horn clause* if it contains at most one positive literal; a CNF formula is a *Horn formula* if it contains only Horn clauses. We let the *degree* of a CNF formula φ be the maximum number of times that any variable $x \in \text{Var}(\varphi)$ occurs in φ. We use the standard notion of *(truth) assignments* $\alpha : \text{Var}(\varphi) \rightarrow \{0, 1\}$ for Boolean formulas and *truth* of a formula under such an assignment. We let SAT denote the problem of deciding whether a given propositional formula is satisfiable, and we let UNSAT denote its co-problem, i.e., deciding whether a given formula is unsatisfiable. We say that a propositional formula is *doubly-negated* if it is of the form $\neg\neg\psi$. For every propositional formula φ, we let $\sim\varphi$ denote the *complement* of φ, i.e., $\sim\varphi = \neg\varphi$ if φ is not of the form $\neg\psi$, and $\sim\varphi = \psi$ if φ is of the form $\neg\psi$.

An *agenda* is a finite nonempty set Φ of formulas that does not contain any doubly-negated formulas and that is closed under complementation. Moreover, if $\Phi = \{\varphi_1, \dots, \varphi_n, \neg\varphi_1, \dots, \neg\varphi_n\}$ is an agenda, then we let $[\Phi] = \{\varphi_1, \dots, \varphi_n\}$ denote the *pre-agenda* associated to the agenda Φ. A *judgment set* J for an agenda Φ is a subset $J \subseteq \Phi$. We call a judgment set J *complete* if $\varphi \in J$ or $\sim\varphi \in J$ for all $\varphi \in \Phi$; we call it *complement-free* if for all $\varphi \in \Phi$ it is not the case that both φ and $\sim\varphi$ are in J; and we call it *consistent* if there exists an assignment that makes all formulas in J true. Let $\mathcal{J}(\Phi)$ denote the set of all complete and consistent subsets of Φ. Let $\mathcal{N}$ be a set of agents, with $|\mathcal{N}| = n$. We call a sequence $\boldsymbol{J} \in \mathcal{J}(\Phi)^n$ of complete and consistent subsets a *profile*. A (resolute) *judgment aggregation procedure* for the agenda Φ and the set of individuals $\mathcal{N}$ is a function $F : \mathcal{J}(\Phi)^n \rightarrow 2^\Phi$. An example is the *majority rule* F^{maj}, where $\varphi \in F^{\text{maj}}(\boldsymbol{J})$ if and only if φ occurs in the majority of judgment sets in $\boldsymbol{J}$, for all $\varphi \in \Phi$. We call F *complete*, *complement-free* and *consistent*, if $F(\boldsymbol{J})$ is complete, complement-free and consistent, respectively, for every $\boldsymbol{J} \in \mathcal{J}(\Phi)^n$. An agenda Φ is *safe* with respect to a class of aggregation procedures $\mathcal{F}$, if every procedure in $\mathcal{F}$ is consistent when applied to profiles of judgment sets over Φ. We say that an agenda Φ satisfies the *median property (MP)* if every inconsistent subset of Φ has itself an inconsistent subset of size at most 2. An agenda Φ is safe for the majority rule if and only if Φ satisfies the MP [20, 36]. There exist similar properties that characterize agenda safety for other aggregation procedures [20].

As an example, we consider the *discursive dilemma*, which concerns an agenda that is not safe for the majority rule. Consider the agenda $\Phi_{\text{dd}} = \{p, \neg p, q, \neg q, (p \rightarrow q), \neg(p \rightarrow q)\}$. Moreover, consider the profile $\boldsymbol{J} = (J_1, J_2, J_3)$, where $J_1 = \{p, q, (p \rightarrow q)\}$, $J_2 = \{p, \neg q, \neg(p \rightarrow q)\}$, and $J_3 = \{\neg p, \neg q, (p \rightarrow q)\}$. Each of these judgment sets are consistent. However, $F^{\text{maj}}(\boldsymbol{J}) = \{p, \neg q, (p \rightarrow q)\}$ is inconsistent. In other words, Φ_{dd} is not safe for the majority rule. Also, Φ_{dd} does not satisfy the MP, as it contains the subset $F^{\text{maj}}(\boldsymbol{J}) \subseteq \Phi$ that is inconsistent, but that itself contains no inconsistent subset of size 2. Intuitively, for each agenda that does not satisfy the MP, a similar discursive dilemma can be constructed, where the majority rule is forced to in-

Parameter	Complexity
maximum formula size (ℓ)	para-Π_2^{P}-complete, even when restricted to 2CNF $\cap$ HORN (Proposition 3)
maximum variable degree (d)	para-Π_2^{P}-complete, even when restricted to 2CNF $\cap$ HORN (Proposition 3)
agenda size	solvable in fpt-time with $f(k)$ many SAT calls, with $f(k) = 2^{O(k)}$ (Theorem 1) and $f(k) = \Omega(\log k)$ (Theorem 2)
counterexample size	$\forall^k\exists^*$-hard (Theorem 3)
formula primal treewidth	fixed-parameter tractable (Proposition 6)
clausal primal and incidence treewidth	para-co-NP-complete (Propositions 7 and 8)
formula incidence treewidth	para-Π_2^{P}-complete (Proposition 9)

Table 1: Complexity results for different parameterizations of agenda safety.

clude an inconsistent subset (of size larger than 2), whereas the individual profiles remain consistent.

In this paper, we consider several parameterizations of the following decision problem MAJ-AS, which is known to be Π_2^{P}-complete [20]. The problem MAJ-AS consists of, given an agenda Φ, deciding whether Φ is safe for the majority rule. For our results, we will use the fact that deciding safety of an agenda Φ for the majority rule is equivalent to checking whether Φ satisfies the median property. In fact, the technical details behind our results involve only this alternative characterization.

The Boolean and Polynomial Hierarchies.

There are many natural decision problems that are apparently not contained in the classical complexity classes P or NP. The *Boolean Hierarchy* (BH) [12, 13, 31] consists of a hierarchy of complexity classes BH_i for all $i \geq 1$. Each class BH_i can be characterized as the class of problems that can be reduced in polynomial time to the problem BH_i-SAT, which is defined inductively as follows. The problem BH_1-SAT consists of all sequences (φ), where φ is a satisfiable propositional formula. For even $i \geq 2$, the problem BH_i-SAT consists of all sequences $(\varphi_1, \dots, \varphi_i)$ of propositional formulas such that both $(\varphi_1, \dots, \varphi_{i-1}) \in \mathrm{BH}_{(i-1)}$-SAT and φ_i is unsatisfiable. For odd $i \geq 2$, the problem BH_i-SAT consists of all sequences $(\varphi_1, \dots, \varphi_i)$ of propositional formulas such that $(\varphi_1, \dots, \varphi_{i-1}) \in \mathrm{BH}_{(i-1)}$-SAT or φ_i is satisfiable. The class BH_2 is also denoted by DP, and the problem BH_2-SAT is also denoted by SAT-UNSAT.

The *Polynomial Hierarchy* (PH) [35, 38, 43, 45] consists of a hierarchy of complexity classes, including the classes Σ_i^{p}, for all $i \geq 0$. The class Σ_2^{p} already contains the entire BH. We give a characterization of these classes based on the satisfiability problem of various classes of quantified Boolean formulas. A *(prenex) quantified Boolean formula* is a formula of the form $Q_1X_1Q_2X_2 \dots Q_mX_m\psi$, where each Q_i is either $\forall$ or $\exists$, the X_i are disjoint sets of propositional variables, and ψ is a Boolean formula over the variables in $\bigcup_{i=1}^m X_i$. The quantifier-free part of such formulas is called the *matrix* of the formula. Truth of such formulas is defined in the usual way. We let $\psi[\alpha]$ denote the formula obtained from ψ by instantiation variables by their truth values given by a (partial) truth assignment α. For each $i \geq 1$ we define the decision problem QSAT_i, where the problem is to decide whether a given quantified Boolean formula $\varphi = \exists X_1\forall X_2\exists X_3 \dots Q_iX_i\psi$ is true, where Q_i is a universal quantifier if i is even and an existential quantifier if i is odd. For each nonnegative integer $i \geq 0$, the complexity class Σ_i^{p} is the class of problems that can be reduced to QSAT_i in polynomial time [43, 45]. The Σ_i^{p}-hardness of QSAT_i holds already when the matrix of the input formula is restricted to 3CNF for odd i, and restricted to 3DNF for even i. Note that $\Sigma_0^{\mathrm{p}} = \mathrm{P}$, and that $\Sigma_1^{\mathrm{p}} = \mathrm{NP}$. For each $i \geq 1$, the class Π_i^{P} is defined as co-Σ_i^{p}.

Parameterized Complexity.

We introduce some core notions from parameterized complexity theory that we will use in this paper. For an in-depth treatment we refer to other sources [18, 19, 22, 28, 37]. A *parameterized problem* L is a subset of $\Sigma^* \times \mathbb{N}$ for some finite alphabet Σ. For an instance $(I, k) \in \Sigma^* \times \mathbb{N}$, we call I the *main part* and k the *parameter*. The following generalization of polynomial time computability is commonly regarded as the tractability notion of parameterized complexity theory. A parameterized problem L is *fixed-parameter tractable* if there exists a computable function f and a constant c such that there exists an algorithm that decides whether $(I, k) \in L$ in time $O(f(k)||I||^c)$, where $||I||$ denotes the size of I. Such an algorithm is called an *fpt-algorithm*, and this amount of time is called *fpt-time*. FPT is the class of all fixed-parameter tractable parameterized decision problems. If the parameter is constant, then fpt-algorithms run in polynomial time where the order of the polynomial is independent of the parameter. This provides a good scalability in the parameter in contrast to running times of the form $||I||^k$, which are also polynomial for fixed k, but are already impractical for, say, $k > 3$. By XP we denote the class of all problems L for which it can be decided whether $(I, k) \in L$ in time $O(||I||^{f(k)})$, for some fixed computable function f.

Let $L \subseteq \Sigma^* \times \mathbb{N}$ and $L' \subseteq (\Sigma')^* \times \mathbb{N}$ be two parameterized problems. An *fpt-reduction* from L to L' is a mapping $R : \Sigma^* \times \mathbb{N} \to (\Sigma')^* \times \mathbb{N}$ from instances of L to instances of L' such that there exist some computable function $g : \mathbb{N} \to \mathbb{N}$ such that for all $(I, k) \in \Sigma^* \times \mathbb{N}$: (i) (I, k) is a yes-instance of L if and only if $(I', k') = R(I, k)$ is a yes-instance of L', (ii) $k' \leq g(k)$, and (iii) R is computable in fpt-time.

Let C be a classical complexity class, e.g., NP. The parameterized complexity class para-C is then defined as the class of all parameterized problems $L \subseteq \Sigma^* \times \mathbb{N}$, for some finite alphabet Σ, for which there exists an alphabet Π, a computable function $f : \mathbb{N} \to \Pi^*$, and a problem $P \subseteq \Sigma^* \times \Pi^*$ such that $P \in C$ and for all instances $(x, k) \in \Sigma^* \times \mathbb{N}$ of L we have that $(x, k) \in L$ if and only if $(x, f(k)) \in P$. Intuitively, the class para-C consists of all problems that are in C after a precomputation that only involves the parameter [21].

In particular, the class para-NP contains those parameterized problems that can be fpt-reduced to a single instance of SAT. Another class containing problems that can be considered fpt-reducible to SAT is the class para-DP, based on the classical complexity class DP $= \{ L_1 \cap L_2 : L_1 \in \text{NP}, L_2 \in \text{co-NP} \}$. An instance of a parameterized problem in para-DP can be solved in fpt-time by firstly reducing it to an instance of the problem SAT-UNSAT $= \{ (\varphi_1, \varphi_2) : \varphi_1 \in \text{SAT}, \varphi_2 \in \text{UNSAT} \}$, and then solving this resulting instance by invoking a SAT oracle twice.

In addition to many-one fpt-reductions to SAT, we are also interested in Turing fpt-reductions. A *Turing fpt-reduction from a problem P to SAT* is an fpt-algorithm that has access to a SAT oracle and that decides P. We are mainly interested in fpt-algorithms that only use a small number of queries to the SAT oracle (*SAT calls*). We let $\text{FPT}^{\text{NP}[f(k)]}$ denote the class of all parameterized problems P for which there exists an fpt-algorithm that decides if $(x, k) \in P$ by using at most $f(k)$ many SAT calls, for some computable function f.

The notion of para-Σ_2^{p}-hardness can be employed to provide evidence against the existence of fpt-reductions to SAT. However, for many interesting parameterized problems for which we want to investigate the (non-)existence of fpt-reductions to SAT, hardness for para-Σ_2^{p} cannot be used. The class para-Σ_2^{p} contains problems that cannot be reduced to SAT in polynomial time if the parameter value is a constant (unless the Polynomial Hierarchy collapses at the first level), i.e., problems in para-Σ_2^{p} do not allow an xp-reduction to SAT. Since many problems we are interested in do allow such xp-reductions to SAT, it is unlikely that these problems can be shown to be hard for the complexity class para-Σ_2^{p}.

Recent work in parameterized complexity theory has resulted in complexity classes that can be used to provide evidence for the non-existence of fpt-reductions to SAT also for problems that do allow an xp-reduction to SAT [26, 28]. The parameterized complexity class $\forall^k\exists^*$ consists of all parameterized problems that can be fpt-reduced to the following variant of quantified Boolean satisfiability that is based on truth assignments of restricted (Hamming) weight (the Hamming weight of an assignment is the number of variables that it assigns to 1). The problem $\forall^k\exists^*$-WSAT consists of deciding, for a given quantified Boolean formula $\varphi = \forall X.\exists Y.\psi$ and a given integer k, whether for all truth assignments α to X of weight k there exists a truth assignment β to Y such that the assignment $\alpha \cup \beta$ satisfies ψ. The parameter is k.

For any problem in $\forall^k\exists^*$ there exists an xp-reduction to SAT. However, there is evidence that problems that are hard for $\forall^k\exists^*$ do not allow an fpt-reduction to SAT [26, 28]. Many natural parameterized problems from various domains are complete for the class $\forall^k\exists^*$, and for none of them an fpt-reduction to SAT has been found [26]. If there exists an fpt-reduction to SAT for any $\forall^k\exists^*$-complete problem then this is the case for all $\forall^k\exists^*$-complete problems. For an overview of parameterized complexity classes that are relevant to the results in this paper, we refer to Figure 1 (for a definition of the classes W[1], co-W[1] and Δ_2^{p}, referred to in this figure, we refer to other sources [18, 19, 22]). For a more detailed discussion on this topic, we refer to previous work in parameterized complexity [26, 28].

Treewidth.

Let $G = (V, E)$ be a graph. A tree decomposition of G is a pair $(\mathcal{T}, (B_t)_{t \in T})$ where $\mathcal{T} = (T, F)$ is a rooted tree and $(B_t)_{t \in T}$ is a family of subsets of V such that (1) for every $v \in V$, the set $B^{-1}(v) = \{ t \in T : v \in B_t \}$ is nonempty and connected in $\mathcal{T}$; and (2) for every edge $\{v, w\} \in E$, there is a $t \in T$ such that $v, w \in B_t$. The *width* of the decomposition is the number $\max\{ |B_t| : t \in T \} - 1$. The *treewidth* of G is the minimum of the widths of all tree decompositions of G. Let G be a graph and k a positive integer. There is an algorithm that computes a tree decomposition of G of width k, if it exists, and fails otherwise; this algorithm runs in linear time for fixed k [9].

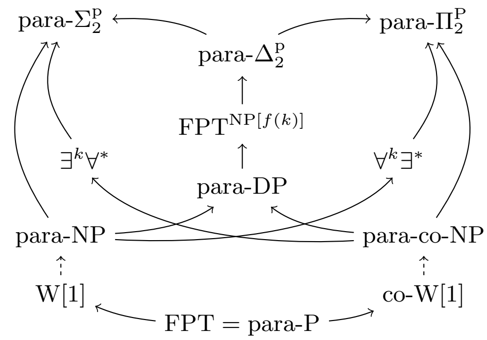

Figure 1: Parameterized complexity classes relevant to the results in this paper. Arrows indicate inclusion relations.

Treewidth is often used as a parameter to represent the amount of structure present in CNF formulas. There are several ways of associating treewidth to a CNF formula. Two of the most common ways are the primal and incidence treewidth. Let φ be a CNF formula. The *primal graph* of φ has as vertices the variables occurring in φ, and two variables are connected by an edge if there exists a clause in which they both occur. The *incidence graph* of φ is a bipartite graph whose vertices consist of (1) the variables occurring in φ and (2) the clauses of φ. A variable x is connected by an edge to a clause c if x occurs in c. The *primal treewidth* of φ is the treewidth of its primal graph, and the *incidence treewidth* of φ is the treewidth of its incidence graph.

3. COMPLEXITY RESULTS

We start with showing that we can restrict our attention to agendas containing only formulas in CNF. We show how to transform any agenda Φ to an agenda Φ' of size polynomial in the size of Φ, containing only formulas in CNF (and their negations), that is safe if and only if Φ is safe. For this, we will need the following proposition, that can be shown by using the well-known Tseitin transformation [44] (we omit its straightforward proof).

PROPOSITION 1. *Let Φ be an agenda with $[\Phi] = \{\varphi_1, \dots, \varphi_n\}$. We can construct in polynomial time an agenda Φ' with $[\Phi'] = \{\varphi'_1, \dots, \varphi'_n\}$ such that each φ'_i is in CNF and any subset $\Psi = \{\varphi_{i_1}, \dots, \varphi_{i_{m_1}}, \neg\varphi_{j_1}, \dots, \neg\varphi_{j_{m_2}}\}$ of Φ is consistent if and only if $\Psi' = \{\varphi'_{i_1}, \dots, \varphi'_{i_m}, \neg\varphi'_{j_1}, \dots, \neg\varphi'_{j_{m_2}}\}$ is consistent.*

PROOF (SKETCH). Let Φ be an agenda with $[\Phi] = \{\varphi_1, \dots, \varphi_n\}$. By using the well-known Tseitin transformation [44], we can transform each φ_i in linear time to a CNF formula φ'_i such that $\text{Var}(\varphi'_i) \supseteq \text{Var}(\varphi_i)$ and for each truth assignment $\alpha : \text{Var}(\varphi_i) \rightarrow \{0,1\}$ we have that α satisfies φ_i if and only if there exists an assignment $\beta : (\text{Var}(\varphi'_i)\backslash\text{Var}(\varphi_i)) \rightarrow \{0,1\}$ such that the assignment $\alpha \cup \beta$ satisfies φ'_i. Because we can introduce fresh variables for constructing each φ'_i, we can assume without loss of generality that for each $1 \leq i < i' \leq n$ it is the case that $(\text{Var}(\varphi'_i)\backslash\text{Var}(\varphi_i)) \cap (\text{Var}(\varphi'_{i'})\backslash\text{Var}(\varphi_{i'})) = \emptyset$. Let $\Psi = \{\varphi_{i_1}, \dots, \varphi_{i_{m_1}}, \neg\varphi_{j_1}, \dots, \neg\varphi_{j_{m_2}}\}$ be an arbitrary subset of Φ. We claim that Ψ is consistent if and only if $\Psi' = \{\varphi'_{i_1}, \dots, \varphi'_{i_{m_1}}, \neg\varphi'_{j_1}, \dots, \neg\varphi'_{j_{m_2}}\}$ is consistent. $\square$

Intuitively, the above results show that, using additional auxiliary variables, each agenda can be rewritten into another agenda that contains only formulas in CNF (or their negation) that are equivalent (with respect to satisfiability) to the formulas in the original agenda.

3.1 Simple Syntactic Restrictions

We consider the following parameterizations of the agenda safety problem that correspond to syntactic restrictions on the agenda Φ. We parameterize on the size of formulas $\varphi \in \Phi$, on the maximum number of times any variable occurs in Φ (i.e., the degree of Φ), and on the number of formulas occurring in Φ. Concretely, we consider the parameterized problems MAJ-AS(formula-size), where the parameter is $\ell = \max\{\, ||\varphi|| : \varphi \in \Phi \,\}$; MAJ-AS(degree), where the parameter is the degree d of Φ; MAJ-AS(degree + formula size), where the parameter is $\ell + d$; and MAJ-AS(ag.-size), where the parameter is $|\Phi|$. Here we define the *degree* of an agenda Φ to be the maximum number of times that any variable $x \in \text{Var}(\Phi)$ occurs in $[\Phi]$, i.e., $\max_{x\in\text{Var}(\Phi)}(\sum_{\varphi\in[\Phi]} \text{occ}(x,\varphi))$, where $\text{occ}(x,\varphi)$ denotes the number of times that x occurs in φ.

The assumption that the size of formulas in an agenda is small corresponds to the expectation that the separate statements that the individuals are judging are in a sense atomic, and therefore of bounded size. The assumption that the degree of an agenda is small corresponds to the expectation that each proposition that occurs in the statements to be judged occurs only a small number of times. The assumption that the number of formulas in the agenda is small is based on the fact that the individuals need to form an opinion on all formulas in the agenda.

Agendas with Small Formulas and Small Degree.

We start by showing that parameterizing on (the sum of) the maximum formula size and the degree of the agenda Φ does not decrease the complexity of deciding whether the agenda is safe, even when (the pre-agenda associated to) Φ contains only formulas in 2CNF $\cap$ HORN. Intuitively, these restrictions on the form and size of the formulas in the agenda do not rule out the complex interactions between the formulas in the agenda that involve many formulas simultaneously, and that give rise to the Π^{P}_2-hardness of the problem.

PROPOSITION 2. MAJ-AS(formula-size) *is* para-Π^{P}_2-*complete.*

PROOF (SKETCH). Membership in para-Π^{P}_2 follows from the Π^{P}_2-membership of MAJ-AS. We show para-Π^{P}_2-hardness by giving a polynomial-time reduction from $\forall\exists$-SAT(3CNF) to the problem $\{\, x : (x,c) \in \text{MAJ-AS(formula-size)} \,\}$, where c is bounded by the size of formulas of the form $\neg((\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge \neg z)$. This reduction is a modified variant of a reduction given by Endriss et al. [20, Lemma 11]. Let $\varphi = \forall X.\exists Y.\psi$ be an instance of $\forall\exists$-SAT, where $\psi = c_1 \wedge \dots \wedge c_m$ is in 3CNF, and where $X = \{x_1, \dots, x_m\}$. We may assume without loss of generality that none of the c_i is a unit clause. We construct the agenda $\Phi = \{x_1, \neg x_1, \dots, x_n, \neg x_n, (c_1 \wedge \neg z_1), \neg(c_1 \wedge \neg z_1), \dots, (c_m \wedge \neg z_m), \neg(c_m \wedge \neg z_m)\}$, where $Z = \{z_1, \dots, z_m\}$ is a set of fresh variables. We claim that Φ satisfies the median property if and only if φ is true. $\square$

Next, using the following technical lemma (whose straightforward proof is omitted), and the reduction given in the proof of Proposition 2, we get para-Π^{P}_2-completeness of MAJ-AS(degree + formula size). The hardness result holds even when we restrict the formulas to be in HORN $\cap$ 2CNF.

LEMMA 1. *The problem* $\forall\exists$-SAT(3CNF) *is* Π^{P}_2-*hard even when restricted to instances* $\varphi = \forall X.\exists Y.\psi$ *where each* $x \in X$ *occurs at most* 2 *times in* ψ *and each* $y \in Y$ *occurs at most* 3 *times in* ψ.

PROPOSITION 3. MAJ-AS(degree + formula size) *is* para-Π^{P}_2-*hard even when restricted to agendas* Φ *such that all formulas* $\varphi \in [\Phi]$ *are in* HORN $\cap$ 2CNF.

PROOF. We consider the reduction used to show Proposition 2. The agenda Φ that we constructed contains only formulas of the form x_i or their negation, and formulas of the form $(c_i \wedge \neg z_i)$, where c_i is a clause, or their negation. Clearly, the formulas x_i and $\neg x_i$ are (equivalent to formulas) in HORN $\cap$ 2CNF. It suffices to show that each formula $\varphi \in \Phi$ with $\varphi = (c_i \wedge \neg z_i)$ is equivalent to a formula $\varphi' \in$ HORN $\cap$ 2CNF. Let $c_i = (l^i_1 \vee l^i_2 \vee l^i_3)$. Observe that $(c_i \wedge \neg z_i) = ((l^i_1 \vee l^i_2 \vee l^i_3) \wedge \neg z_i) \equiv (l^i_1 \vee \neg z_i) \wedge (l^i_2 \vee \neg z_i) \wedge (l^i_3 \vee \neg z_i)$. Thus, we can construct Φ in such a way that $[\Phi]$ contains only formulas in HORN $\cap$ 2CNF. $\square$

Agendas with Few Formulas.

Next, we parameterize the agenda safety problem on the number of formulas occurring in the agenda. We will show that instances (x,k) of the problem MAJ-AS(ag.-size) can be solved by an fpt-algorithm that uses $f(k)$ many SAT calls. Intuitively, the fpt-algorithm that we construct will exploit the fact that the agenda only contains few formulas, by considering all possible inconsistent subsets of the agenda, and using a SAT solver to verify that these all have an inconsistent subset of size at most 2. In particular, we will prove the following result.

THEOREM 1. *There exists an algorithm that decides* MAJ-AS(ag.-size) *in fpt-time using at most* $2^{O(k)}$ *SAT calls, where* k *is the parameter value.*

Moreover, we give evidence that this is the best one can do, i.e., there exists no fpt-algorithm that uses a significantly smaller number of SAT calls, assuming some widely believed complexity-theoretic assumptions (Theorem 2). In order to perform our lower-bound analysis, we will consider the parameterized complexity class $\text{FPT}^{\text{NP}[f(k)]}$. We will show that MAJ-AS(ag.-size) is complete for this class.

We begin with considering the following auxiliary problem BH(level)-SAT, and showing that it is $\text{FPT}^{\text{NP}[f(k)]}$-complete. Given a positive integer k and a sequence $\overline{\varphi} = (\varphi_1, \dots, \varphi_k)$ of propositional formulas, the problem is to decide whether $\overline{\varphi} \in \text{BH}_k$-SAT. The parameter is k.

Lemma 2. BH(level)-Sat *is* $\mathrm{FPT}^{\mathrm{NP}[f(k)]}$*-complete.*

Proof (sketch). We first show membership. Let (x, k) be an instance of BH(level)-Sat, where $x = (\varphi_1, \dots, \varphi_{k'})$. Then, for each $1 \leq i \leq k$, it decides whether φ_i is satisfiable by a single SAT call. Since (x, k) corresponds to a Boolean combination of statements concerning the satisfiability of the formulas φ_i, the algorithm can then decide in fpt-time whether $(x, k) \in$ BH(level)-Sat.

We can prove hardness by showing the following. Let P be a parameterized problem and let A be an algorithm that decides P in fpt-time using at most $g(k)$ many SAT calls, where k is the parameter value and g is some computable function. Then there are at most $2^{g(k)}$ possible sequences of answers to these calls, that lead the algorithm to accept the input. We can encode the check whether a particular such sequence corresponds to the correct answers to the SAT calls, as a single instance of SAT-UNSAT. Then, the disjunction of all these instances can be encoded as a single instance (x', k') of BH(level)-Sat, where $k' \leq 2^{g(k)+1}$ [12]. We omit a full detailed proof of this claim. $\square$

We now use this completeness result to show the upper bound on the number of SAT calls needed to solve Maj-AS(ag.-size).

Proof of Theorem 1. As a first step, we provide an fpt-algorithm that takes an instance Φ of Maj-AS(ag.-size) with $|\Phi| = k$ and produces $f(k)$ many instances $x_1, \dots, x_{f(k)}$ of co-SAT-UNSAT such that $\Phi \in$ Maj-AS(ag.-size) if and only if $\{x_1, \dots, x_{f(k)}\} \subseteq$ co-SAT-UNSAT. Let Φ be an agenda with $[\Phi] = \{\varphi_1, \dots, \varphi_k\}$. Let C denote the set of all complement-free subagendas $\Phi' \subseteq \Phi$ that are of size at least 3. Clearly, $|C| = 2^{O(k)}$. We know that Φ satisfies the MP if and only if for all $\Phi' \in C$ holds that either (1) Φ' is satisfiable, or (2) there exists some $\Phi'' \subseteq \Phi'$ of size 2 that is unsatisfiable.

Firstly, for each $\Phi' = \{\psi_1, \dots, \psi_\ell\} \in C$, we construct an instance $I(\Phi') = (\psi_1, \psi_2)$ of co-SAT-UNSAT such that $(\psi_1, \psi_2) \in$ co-SAT-UNSAT if and only if either (1) Φ' is satisfiable or (2) there exists some $\Phi'' \subseteq \Phi'$ of size 2 that is unsatisfiable. For any $1 \leq i < j \leq \ell$ and any propositional formula φ, we let $\varphi^{(i,j)}$ denote a copy of φ where each variable $x \in \mathrm{Var}(\varphi)$ is replaced with a copy $x^{(i,j)}$ indexed by the pair (i, j). We define $\psi_1 = \bigwedge_{\varphi \in \Phi'} \varphi$, and $\psi_2 = \bigwedge_{1 \leq i < j \leq \ell} (\psi_i^{(i,j)} \wedge \psi_j^{(i,j)})$. It is straightforward to verify that $I(\Phi')$ satisfies the required properties.

We now straightforwardly get that $\Phi \in$ Maj-AS(ag.-size) if and only if $\{\, I(\Phi') : \Phi' \in C \,\} \subseteq$ co-SAT-UNSAT. Also, we know that $|C| = f(k) = 2^{O(k)}$ for a suitable computable function f. We know that the conjunction of $f(k)$ many instances of co-SAT-UNSAT can be reduced in polynomial time to an instance of co-$\mathrm{BH}_{2f(k)}$-Sat [12]. By Lemma 2, this implies that Maj-AS(ag.-size) is in $\mathrm{FPT}^{\mathrm{NP}[f(k)]}$. Moreover, the algorithm that witnesses this decides Maj-AS(ag.-size) in time $O(n \cdot 2^k)$ by making $O(2^k)$ many queries to a SAT solver consisting of formulas of size $O(n \cdot k^2)$, where n is the input size and k is the parameter value. $\square$

Next, we will pursue the lower bound. We start with identifying an easier hardness result, which we will then extend to a hardness result for the class $\mathrm{FPT}^{\mathrm{NP}[f(k)]}$.

Lemma 3. Maj-AS(ag.-size) *is* para-co-DP*-hard.*

Proof (sketch). We prove hardness for para-co-DP by giving a polynomial-time reduction from SAT-UNSAT to co-Maj-AS, such that the resulting instance is an agenda of constant size. Let (φ_1, φ_2) be an instance of SAT-UNSAT. We construct the agenda Φ with $[\Phi] = \{\psi_1, \psi_2, \psi_3\}$ by letting $\psi_1 = r_1 \wedge p_1 \wedge \varphi_1$, $\psi_2 = r_2 \wedge p_2$, and $\psi_3 = r_3 \wedge ((p_1 \wedge p_2) \rightarrow \varphi_2)$, where $\{r_1, r_2, r_3, p_1, p_2\}$ are distinct fresh variables not occurring in φ_1 nor in φ_2. We claim that Φ does not satisfy the MP if and only if $(\varphi_1, \varphi_2) \in$ SAT-UNSAT. $\square$

Proposition 4. Maj-AS(ag.-size) *is* $\mathrm{FPT}^{\mathrm{NP}[f(k)]}$*-hard.*

Proof. We give an fpt-reduction from BH(level)-Sat to co-Maj-AS(ag.-size). For the sake of simplicity, we assume that $k \geq 2$ is even. Let the sequence $(\varphi_1, \dots, \varphi_k)$ specify an instance of BH(level)-Sat. We know that we can construct in polynomial time a sequence of formulas $(\varphi_1, \psi_1, \dots, \varphi_\ell, \psi_\ell)$, where $\ell = k/2$, such that $(\varphi_1, \dots, \varphi_k) \in \mathrm{BH}_k$-Sat if and only if for some $1 \leq i \leq \ell$ it holds that $(\chi_i, \psi_i) \in \mathrm{BH}_2$-Sat = SAT-UNSAT [12].

Now, for each $1 \leq i \leq \ell$, we can use the reduction in the proof of Lemma 3 to construct in polynomial time an agenda Φ_i of constant size such that Φ_i does not satisfy the median property if and only if $(\chi_i, \psi_i) \in$ SAT-UNSAT. Moreover, we can ensure that the agendas Φ_i are variable-disjoint. We now construct the agenda $\Phi = \bigcup_{1 \leq i \leq \ell} \Phi_i$. We claim that Φ does not satisfy the median property if and only if $(\chi_i, \psi_i) \in$ SAT-UNSAT for some $1 \leq i \leq \ell$. We know this latter condition holds if and only if our original instance $(\varphi_1, \dots, \varphi_k) \in \mathrm{BH}_k$-Sat. Moreover, since $|\Phi| = O(k)$, we obtain a correct fpt-reduction. $\square$

We will now use the $\mathrm{FPT}^{\mathrm{NP}[f(k)]}$-hardness of Maj-AS-(ag.-size), to obtain lower bounds on the number of SAT calls needed to solve Maj-AS(ag.-size).

Proposition 5. *Let P be any* $\mathrm{FPT}^{\mathrm{NP}[f(k)]}$*-hard problem. Then P is not solvable by an fpt-algorithm that uses only $O(1)$ many SAT calls, unless the PH collapses.*

Proof. Assume that P is solvable by an fpt-algorithm that uses only c many SAT calls, where c is a constant. We will show that the PH collapses. Since P is $\mathrm{FPT}^{\mathrm{NP}[f(k)]}$-hard, we know that there exists an fpt-reduction R_1 from BH(level)-Sat to P. Then, by (the proof of) Lemma 2, there exists an fpt-reduction R_2 from P to BH(level)-Sat, that reduces any instance (x', k') of P to an instance (x'', k'') of BH(level)-Sat, where $k'' \leq 2^{c+1}$. Then, the composition R of R_1 and R_2 is an fpt-reduction from BH(level)-Sat to itself such that any instance (x, k) of BH(level)-Sat is reduced to an equivalent instance (x'', k'') of BH(level)-Sat, where $k'' \leq 2^{c+1}$. We can straightforwardly modify this reduction to always produce an instance $(x'', 2^{c+1})$ of BH(level)-Sat, by adding trivial instances of SAT to the sequence x''.

We now show that the Boolean Hierarchy collapses to the m-th level, where $m = 2^{c+1}$. Let y be an instance of BH_{m+1}-Sat. We can then see the reduction R as a polynomial-time reduction from BH_{m+1}-Sat to BH_m-Sat: the fpt-reduction R runs in time $f(k) \cdot n^{O(1)}$, and since $k = m+1$ is a constant, the factor $f(k)$ is constant. From this we can conclude that $\mathrm{BH}_m = \mathrm{BH}_{m+1}$. Thus, the BH collapses, and consequently the PH collapses [13, 31]. $\square$

The above lower bound holds for any $\mathrm{FPT}^{\mathrm{NP}[f(k)]}$-hard problem. We can improve this bound for the particular case of Maj-AS(ag.-size).

THEOREM 2. *Deciding whether* $(x, k) \in$ MAJ-AS(ag.-size) *is not solvable by an fpt-algorithm that uses* $o(\log k)$ *many SAT calls, unless the PH collapses.*

PROOF (SKETCH). The proof is analogous to the proof of Proposition 5. Since we know in addition that there exists an fpt-reduction from MAJ-AS(ag.-size) to BH(level)-SAT that increases the parameter value (only) exponentially, the argument from the proof of Proposition 5 gives us the lower bound on the number of SAT calls. □

3.2 Bounded Treewidth

Another type of structure that the agenda Φ can exhibit is the way in which the formulas $\varphi \in \Phi$ interact with each other. As an extreme example, consider the case of an agenda Φ with $[\Phi] = \{\varphi_1, \dots, \varphi_m\}$, and where all formulas φ_i are variable-disjoint. Clearly, any minimal inconsistent subset of this agenda has size 1, and thus this agenda is safe for the majority rule. In less extreme cases, the formulas of the agenda are allowed to interact (i.e., to have variables in common), but their interaction is structured in a particular way. The type of structured interaction that we consider in this section is the 'tree-likeness' of various graphs representing the interaction between formulas of the agenda, captured by the treewidth of these graphs. Treewidth is commonly used in the parameterized complexity analysis of hard problems in various fields, such as graph theory, Boolean satisfiability, constraint satisfaction, and Knowledge Representation and Reasoning. Recently, it has also been used to obtain fpt-reductions to SAT [27]. Intuitively, one could think of agendas of bounded treewidth as agendas where the propositional variables are divided into a number of (thematic) groups, where the interaction between such groups is tree-like. As an example, one could consider an agenda occurring in a court case, where propositions are grouped according to various claims made by the plaintiff, and where these claims support each other in a tree-shaped structure.

Let Φ be an agenda with $[\Phi] = \{\varphi_1, \dots, \varphi_m\}$, where each φ_i is a CNF formula. We define the following graphs that are intended to capture the interaction between formulas in Φ. The *formula primal graph* $\mathcal{G}^{fp}(\Phi)$ of Φ has as vertices the variables $\text{Var}(\Phi)$ occurring in the agenda, and two variables are connected by an edge if there exists a formula φ_i in which they both occur. The *formula incidence graph* $\mathcal{G}^{fi}(\Phi)$ of Φ is a bipartite graph whose vertices consist of (1) the variables $\text{Var}(\Phi)$ occurring in the agenda and (2) the formulas $\varphi_i \in \Phi$. A variable $x \in \text{Var}(\Phi)$ is connected by an edge with a formula $\varphi_i \in \Phi$ if x occurs in φ_i, i.e., $x \in \text{Var}(\varphi_i)$. The *clausal primal graph* $\mathcal{G}^{fp}(\Phi)$ of Φ has as vertices the variables $\text{Var}(\Phi)$ occurring in the agenda, and two variables are connected by an edge if there exists a formula φ_i and a clause $c \in \varphi_i$ in which they both occur. The *clausal incidence graph* $\mathcal{G}^{fi}(\Phi)$ of Φ is a bipartite graph whose vertices consist of (1) the variables $\text{Var}(\Phi)$ occurring in the agenda and (2) the clauses c occurring in formulas $\varphi_i \in \Phi$. A variable $x \in \text{Var}(\Phi)$ is connected by an edge with a clause c of the formula $\varphi_i \in \Phi$ if x occurs in c, i.e., $x \in \text{Var}(c)$.

Now, we consider the following parameterizations of the problem MAJ-AS. The problem MAJ-AS(f-tw) has as parameter the treewidth of the formula primal graph (the *formula primal treewidth*). The problem MAJ-AS(f-tw*) has as parameter the treewidth of the formula incidence graph (the *formula incidence treewidth*). Similarly, the parameterized problems MAJ-AS(c-tw) and MAJ-AS(c-tw*) have as parameters the treewidth of the clausal primal graph and the clausal incidence graph, respectively.

We show that the presence of tree-like structure in three of these four graphs leads to a reduction in the complexity of the problem MAJ-AS. Concretely, we show that in one case, the problem is fixed-parameter tractable, and in two other cases there exists an fpt-reduction to UNSAT.

PROPOSITION 6. MAJ-AS(f-tw) *is fixed-parameter tractable.*

PROOF (SKETCH). We give a high-level description how to prove this by using Courcelle's Theorem. One can encode the agenda Φ into a relational structure, representing by binary relations which formulas contain which clauses and which variables appear (positively or negatively) in which clauses. Moreover, one can do this in such a way that the treewidth of the Gaifman graph of this structure is at most a constant more than the treewidth of the formula primal graph of the agenda. Then, using these relation symbols, one can express the median property with a fixed monadic second-order logic sentence. The fixed-parameter tractability result then follows by Courcelle's Theorem. □

PROPOSITION 7. MAJ-AS(c-tw) *is* para-co-NP-*complete.*

PROOF. To show membership, it suffices to describe a nondeterministic fpt-algorithm that decides whether an instance is a no-instance of MAJ-AS(c-tw) [21]. The algorithm first guesses a subset $\Phi' \subseteq \Phi$ of the agenda of size > 2. Next, the algorithm verifies whether the formula $\psi = \bigwedge_{\varphi \in \Phi'} \varphi$ is unsatisfiable. Since the primal treewidth of ψ is bounded, this can be done in fpt-time [23, 15]. Next, the algorithm verifies whether for each $\varphi' \in \Phi'$ the formula $\psi' = \bigwedge_{\varphi \in \Phi \setminus \{\varphi'\}} \varphi$ is satisfiable. These are $|\Phi'| \leq |\Phi|$ many checks. Since each formula ψ' has bounded primal treewidth, each of these checks can be done in fpt-time. The algorithm accepts if and only if none of these checks fails, and thus it accepts if and only if there exists a minimal unsatisfiable subagenda of size > 2. Therefore, it checks whether a given instance is a no-instance of MAJ-AS in nondeterministic fpt-time. Hardness can be shown with an fpt-reduction from the parameterized variant of UNSAT where the parameter value is a constant. We omit a proof of hardness. □

PROPOSITION 8. MAJ-AS(c-tw*) *is* para-co-NP-*complete.*

PROOF (SKETCH). To show membership, it suffices to describe a nondeterministic fpt-algorithm that decides whether an instance is a no-instance of MAJ-AS(c-tw) [21]. The algorithm works exactly along the same lines as the algorithm given in the proof of membership for Proposition 7. Since satisfiability of CNF formulas of bounded incidence treewidth is also solvable in fpt-time [40], this algorithm also runs in (nondeterministic) fpt-time. Hardness can be shown similarly to the hardness proof of Proposition 7. We omit a proof of hardness. □

Finally, we show that bounding the formula incidence treewidth does not improve the complexity of the problem MAJ-AS.

PROPOSITION 9. MAJ-AS(f-tw*) *is* para-Π_2^{P}*-complete.*

PROOF. We observe that the Π_2^{P}-hardness proof of MAJ-AS given by Endriss, Grandi and Porello [20, Lemmas 22 and 24] shows that the problem MAJ-AS is already Π_2^{P}-hard for agendas with formula incidence treewidth 1. This implies that MAJ-AS(f-tw*) is para-Π_2^{P}-hard. □

3.3 Small Counterexamples

Another commonly identified "hidden" structure in problem instances is a restriction on the size of counterexamples. Many computational problems ask for the non-existence of a particular counterexample, and many of such problems show a decrease in complexity if attention can be restricted to counterexamples of a particular bounded size only.

One prominent example of a decrease in complexity induced by a restriction on the size of counterexamples is the method of Bounded Model Checking [6, 7]. In a nutshell, model checking is the problem of verifying whether a model of a system meets a given specification. This problem finds applications in a myriad of domains. A commonly used formalization is the problem of deciding whether a given transition systems satisfies a specification given in the form of a linear-time temporal logic (LTL) formula. This variant of the problem is PSPACE-complete (cf. [1, 14]). The problem is equivalent to deciding whether there exists no path (potentially of exponential length) in the transition system that serves as a counterexample to the specification. If the size of such counterexamples to consider is bounded (by an upper bound given in the input), the complexity of the problem decreases to NP [6, 7]. This result has been successfully applied in practice, by implementing algorithms that iteratively search for counterexamples of increasing size (cf. [6]). In the worst-case, there can be a counterexample of exponential size, but in many instances occurring in practice, small counterexamples can be found efficiently this way.

A natural question to investigate is whether we could apply a similar approach to deciding whether an agenda is safe for the majority rule. In order to do so, we would like to get an improvement in the computational complexity for the case where the size of counterexamples is bounded. Therefore, we consider the following parameterized variant MAJ-AS(c.e.-size) of the problem MAJ-AS. The problem consists of deciding, given an agenda Φ, and an integer k, whether every inconsistent subset Φ' of Φ of size k has itself an inconsistent subset of size at most 2? The parameter is k.

Assuming that counterexamples to the MP are small in practice corresponds to the supposition that whenever several statements together imply another statement, this latter statement is already implied by a small number of the former statements. In other words, the interaction between statements is, in a sense, local.

This problem is also related to agenda safety for supermajority rules. A supermajority rule accepts any proposition in the agenda if and only if a certain supermajority of the individuals, specified by a threshold $q \in (\frac{1}{2}, 1]$, accepts the proposition. Such rules always produce consistent outcomes if the threshold is greater than $\frac{k-1}{k}$, where k is the size of the largest minimally inconsistent subagenda (cf. [16, 32]).

Unfortunately, it turns out that this parameterization does not lead to a significant (practically exploitable) improvement in the computational complexity. In order to prove this, we will need the following technical lemma (we omit its straightforward proof).

LEMMA 4. *Let (φ, k) be an instance of $\forall^k\exists^*$-WSAT. In polynomial time, we can construct an equivalent instance (φ', k) of $\forall^k\exists^*$-WSAT such that: (1) for every assignment $\alpha : X \to \{0,1\}$ of weight $m > k$, the formula $\exists Y.\psi[\alpha]$ is false; and (2) for every assignment $\alpha : X \to \{0,1\}$ of weigth $m < k$, the formula $\exists Y.\psi[\alpha]$ is true.*

THEOREM 3. MAJ-AS(c.e.-size) *is* $\forall^k\exists^*$*-complete.*

PROOF (SKETCH). In order to show $\forall^k\exists^*$-hardness, we provide an fpt-reduction from $\forall^k\exists^*$-WSAT to MAJ-AS(c.e.-size). Let (φ, k) be an instance of $\forall^k\exists^*$-WSAT, where $\varphi = \forall X.\exists Y.\psi$ is a quantified Boolean formula, $X = \{x_1, \dots, x_n\}$, and k is a positive integer. We may assume without loss of generality that φ satisfies properties (1) and (2) described in Lemma 4. We define the agenda $\Phi = \{x_1, \neg x_1, \dots, x_n, \neg x_n, (\psi \wedge z), \neg(\psi \wedge z)\}$, where z is a fresh variable. We claim that for all assignments $\alpha : X \to \{0,1\}$ of weight k it is the case that $\exists Y.\psi[\alpha]$ is true if and only if every inconsistent subset Φ' of Φ of size $k+1$ has itself an inconsistent subset of size 2. □

Intuitively, restricting attention to counterexamples of size k, still leaves a search space of $O(n^k)$ many possible counterexamples (where n is the input size). Moreover, since there is no restriction on the agenda, searching this space for a counterexample (or verifying that no such counterexample exists) is computationally hard.

4. CONCLUSION

Our main aim, in this paper, was to argue that the complexity analysis of problems in computational social choice that are 'beyond NP' benefits from a parameterized complexity perspective, aiming at obtaining fpt-reductions to SAT in addition to fixed-parameter tractability results. As a concrete case study to kick-off this line of investigation, we provided a parameterized complexity analysis of the problem of agenda safety for the majority rule in judgment aggregation. We identified several positive cases, in addition to several negative cases. In several positive cases, the safety of the agenda can be decided by reducing the problem to a single SAT instance. In another positive case, we can decide whether the agenda is safe for the majority rule in fpt-time using a small number of SAT calls. Moreover, for this case, we identified lower bounds on the number of SAT calls needed to solve the problem in fpt-time.

We hope that the initial results obtained in this paper prove to be the beginning of a structured parameterized complexity investigation of problems in the field of computational social choice that are located at higher levels of the PH. One concrete direction for further research would be to explicitly develop fpt-reductions to SAT for the cases where this is possible, and to optimize them for practical use. In addition, it would be interesting to study the parameterized complexity of the problem of agenda safety for other judgment aggregation procedures.

Acknowledgments

Ronald de Haan and Stefan Szeider are supported by the European Research Council (ERC), project 239962 (COMPLEX REASON), and the Austrian Science Fund (FWF), project P26200 (Parameterized Compilation).

REFERENCES

[1] C. Baier and J.-P. Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[2] D. Baumeister, F. Brandt, F. A. Fischer, J. Hoffmann, and J. Rothe. The complexity of computing minimal unidirectional covering sets. *Theory Comput. Syst.*, 53(3):467–502, 2013.

[3] D. Baumeister, G. Erdélyi, and J. Rothe. How hard is it to bribe the judges? A study of the complexity of bribery in judgment aggregation. In *Proceedings of the Second International Conference on Algorithmic Decision Theory (ADT 2011), Piscataway, NJ, USA, October 26–28, 2011*, volume 6992 of *Lecture Notes in Computer Science*, pages 1–15. Springer Verlag, 2011.

[4] N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier. Studies in computational aspects of voting – a parameterized complexity perspective. In H. L. Bodlaender, R. Downey, F. V. Fomin, and D. Marx, editors, *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*, pages 318–363. Springer Verlag, 2012.

[5] N. Betzler, J. Guo, and R. Niedermeier. Parameterized computational complexity of Dodgson and Young elections. *Information and Computation*, 208(2):165–177, 2010.

[6] A. Biere. Bounded model checking. In A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 457–481. IOS Press, 2009.

[7] A. Biere, A. Cimatti, E. M. Clarke, and Y. Zhu. Symbolic model checking without BDDs. In R. Cleaveland, editor, *Proceedings of the 5th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 1999*, volume 1579 of *Lecture Notes in Computer Science*, pages 193–207. Springer Verlag, 1999.

[8] A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.

[9] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[10] H. L. Bodlaender, R. Downey, F. V. Fomin, and D. Marx, editors. *The Multivariate Algorithmic Revolution and Beyond – Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*. Springer Verlag, 2012.

[11] S. Bouveret and J. Lang. Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. *J. Artif. Intell. Res.*, 32:525–564, 2008.

[12] J. Cai, T. Gundermann, J. Hartmanis, L. A. Hemachandra, V. Sewelson, K. W. Wagner, and G. Wechsung. The Boolean hierarchy I: Structural properties. *SIAM J. Comput.*, 17(6):1232–1252, 1988.

[13] R. Chang and J. Kadin. The Boolean hierarchy and the polynomial hierarchy: a closer connection. *SIAM J. Comput.*, 25:169–178, 1993.

[14] E. M. Clarke, D. Kroening, J. Ouaknine, and O. Strichman. Completeness and complexity of bounded model checking. In B. Steffen and G. Levi, editors, *Verification, Model Checking, and Abstract Interpretation, 5th International Conference (VMCAI 2004)*, volume 2937 of *Lecture Notes in Computer Science*, pages 85–96. Springer Verlag, 2004.

[15] R. Dechter and J. Pearl. Tree clustering for constraint networks. *Artificial Intelligence*, 38(3):353–366, 1989.

[16] F. Dietrich and C. List. Judgment aggregation by quota rules: Majority voting generalized. *J. of Theoretical Politics*, 19(4):391–424, 2007.

[17] B. Dorn and I. Schlotter. Multivariate complexity analysis of swap bribery. *Algorithmica*, 64(1):126–151, 2012.

[18] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer Verlag, New York, 1999.

[19] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer Verlag, 2013.

[20] U. Endriss, U. Grandi, and D. Porello. Complexity of judgment aggregation. *J. Artif. Intell. Res.*, 45:481–514, 2012.

[21] J. Flum and M. Grohe. Describing parameterized complexity classes. *Information and Computation*, 187(2):291–319, 2003.

[22] J. Flum and M. Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science*. Springer Verlag, Berlin, 2006.

[23] E. C. Freuder. A sufficient condition for backtrack-bounded search. *J. of the ACM*, 29(1):24–32, 1982.

[24] C. P. Gomes, H. Kautz, A. Sabharwal, and B. Selman. Satisfiability solvers. In *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 89–134. Elsevier, 2008.

[25] G. Gottlob, R. Pichler, and F. Wei. Bounded treewidth as a key to tractability of knowledge representation and reasoning. *Artificial Intelligence*, 174(1):105–132, 2010.

[26] R. de Haan and S. Szeider. Compendium of parameterized problems at higher levels of the Polynomial Hierarchy. Technical Report TR14–143, *Electronic Colloquium on Computational Complexity* (ECCC), 2014.

[27] R. de Haan and S. Szeider. Fixed-parameter tractable reductions to SAT. In U. Egly and C. Sinz, editors, *Proceedings of the 17th International Symposium on the Theory and Applications of Satisfiability Testing (SAT 2014) Vienna, Austria, July 14–17, 2014*, volume 8561 of *Lecture Notes in Computer Science*, pages 85–102. Springer Verlag, 2014.

[28] R. de Haan and S. Szeider. The parameterized complexity of reasoning problems beyond NP. In C. Baral, G. De Giacomo, and T. Eiter, editors, *Proceedings of the Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2014), Vienna, Austria, July 20-24, 2014*. AAAI Press, 2014.

[29] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *J. of the ACM*, 44(6):806–825, 1997.

[30] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, 2005.

[31] J. Kadin. The polynomial time hierarchy collapses if the Boolean hierarchy collapses. *SIAM J. Comput.*, 17(6):1263–1282, Dec. 1988.

[32] C. List. The theory of judgment aggregation: an introductory review. *Synthese*, 187(1):179–207, 2012.

[33] C. List and C. Puppe. Judgment aggregation: A survey. In *Handbook of Rational and Social Choice*. Oxford University Press, 2009.

[34] S. Malik and L. Zhang. Boolean satisfiability from theoretical hardness to practical success. *Communications of the ACM*, 52(8):76–82, 2009.

[35] A. R. Meyer and L. J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *SWAT*, pages 125–129. IEEE Computer Soc., 1972.

[36] K. Nehring and C. Puppe. The structure of strategy-proof social choice - part I: General characterization and possibility results on median spaces. *J. of Economic Theory*, 135(1):269–305, 2007.

[37] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, 2006.

[38] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[39] K. A. Sakallah and J. Marques-Silva. Anatomy and empirical evaluation of modern SAT solvers. *Bulletin of the European Association for Theoretical Computer Science*, 103:96–121, 2011.

[40] M. Samer and S. Szeider. Algorithms for propositional model counting. *J. Discrete Algorithms*, 8(1):50–64, 2010.

[41] J. Siekmann and G. Wrightson, editors. *Automation of reasoning. Classical Papers on Computer Science 1967–1970*, volume 2. Springer Verlag, 1983.

[42] M. Slavkovik. *Judgment Aggregation for Multiagent Systems*. PhD thesis, University of Luxembourg, 2012.

[43] L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[44] G. S. Tseitin. Complexity of a derivation in the propositional calculus. *Zap. Nauchn. Sem. Leningrad Otd. Mat. Inst. Akad. Nauk SSSR*, 8:23–41, 1968. English translation reprinted in [41].

[45] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33, 1976.