ECAI 2023
K. Gal et al. (Eds.)

© 2023 The Authors.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/FAIA230285

Efficiently Computing Smallest Agreeable Sets

Robert Bredereck[a], Till Fluschnik[a] and Nimrod Talmon[c]

[a]Institut für Informatik, TU Clausthal, Clausthal-Zellerfeld, Germany
Email: robert.bredereck@tu-clausthal.de, till.fluschnik@tu-clausthal.de
[c]Ben-Gurion University, Be'er Sheva, Israel
Email: talmonn@bgu.ac.il
ORCiD ID: Robert Bredereck https://orcid.org/0000-0002-6303-6276,
Till Fluschnik https://orcid.org/0000-0003-2203-4386, Nimrod Talmon https://orcid.org/0000-0001-7916-0979

Abstract. We study the computational complexity of identifying a small agreeable subset of items. A subset of items is agreeable if every agent does not prefer its complement set. We study settings in which agents either can assign arbitrary utilities to the items; can approve or disapprove the items; or can rank the items (in which case we consider Borda utilities). We prove that deciding whether an agreeable set exists is NP-hard for all variants; and we perform a parameterized analysis regarding the following natural parameters: the number of agents, the number of items, and the upper bound on the size of the agreeable set in question.

1 Introduction

Consider the following illustrative example: Some agents (e.g. kids, sponsors) are in a room (e.g. playroom, gallery) that contains some desirable items (e.g. toys, pieces of art). There is a second room, which initially contains no items. Assuming that each agent will move to the second room only when they prefer the items in that room to those in the first room—*what is the minimum number of items one has to move to the second room to ensure that all agents move to the second room? How can we identify these items?*

The problem that is sketched above is called the Smallest Agreeable Subset (SAS) problem. Manurangsi and Suksompong [29] discuss its role and potential applications as a very basic model in the context of fair division of resources.

Remark 1. While the example above is given mainly for illustration purposes, we note that SAS has other use cases as well. Generally speaking, SAS captures a basic stability property that, when not met by an algorithm that selects a subset of elements, may cause group members to defect. Indeed, the property of preferring the set that is selected to the complement set containing the elements that were not selected relates to envy-freeness in other contexts of resource allocation. More concretely, other application of SAS include organizing a trip with a shared luggage, picking prize bundles for winning teams, or splitting resources for settings with several disjoint communities.

With efficient algorithms in relevant special cases (see our contributions in Section 1.2 for details), finding smallest agreeable subsets may become applicable as subroutine for more complex decision-making processes.

Here we are interested in the computational complexity of SAS; and, hence, consider the following computational problem:

Problem 1. SMALLEST AGREEABLE SUBSET (SAS)

Input: A set $C = \{c_1, \ldots, c_m\}$ of items, a set $A = \{a_1, \ldots, a_n\}$ of agents, each $a \in A$ with some utility function $u_a : 2^C \to \mathbb{N} \cup \{0\}$, and an integer $k \in \mathbb{N}$.

Question: Is there a subset $C' \subseteq C$ with $|C'| \leq k$ such that for every agent $a \in A$ it holds that $u_a(C') \geq u_a(C \setminus C')$?

In this work, we only study additive utility functions, i.e., where

$$u_a(C') = \sum_{c' \in C'} u_a(c') \text{ for every } C' \subseteq C$$

(we slightly abuse notation and overload u_a as we hence only specify $u_a\colon C \to \mathbb{N} \cup \{0\}$). We analyze the parameterized complexity of this problem up to efficient and effective data reduction (problem kernelization) for the presumably most natural parameters, namely the number n of agents, the number m of items, and the size k of the agreeable set. Concretely, we consider several types of agent preferences, all being variants of additive (utility-based) preferences: general additive preferences, approval preferences (each utility value is either one or zero), t-approval and t-veto preferences (approval preferences ensuring that there are exactly t ones or t zeros), and ordinal preferences with Borda utilities. In the case of ordinal preferences with Borda utilities, each agent has an internal linear ordering of the items and assigns utility $m-1$ to the most preferred, utility $m-2$ to the second most preferred, and so on.

1.1 Related Work

Based on the fairness notion of envy-freeness, agreeable subsets (e.g. sets of items that are preferred towards their complements) have been considered in different variants in the context of fair division [2, 9, 10]. The problem of finding minimum size agreeable subsets (SAS) was formalized by Suksompong [33], who showed tight upper bounds for the minimum size of agreeable subsets for monotonic preferences as well as an approximation algorithm for two or three players for responsive preferences. Manurangsi and Suksompong [29] developed efficient approximation algorithms for ordinal preferences over single items, both for the value oracle preference model and for additive preferences. Particularly relevant for our work, they showed that computing a minimum-size agreeable subset is strongly NP-hard and hard to approximate within a factor of $(1-\delta)\ln n$ for additive preferences, making their $\ln n$ approximation algorithm for additive preferences essentially tight. Gourvès [26] introduced an extension where the agreeable subset must satisfy extra matroidal constraints. They show worst-case upper bounds on the size of agreeable subsets, and approximation algorithms (in particular, they show a constant-factor approximation for two agents having additive preferences).

To the best of our knowledge, no parameterized complexity results are currently known for SAS. Yet, several multivariate complexity studies [19] have recently been performed in the context of computing envy-free allocations, where, in contrast to SAS, each agent gets its own bundle of items and compares its value to the bundles of the other agents. The first work in this context is the work of Bliem et al. [7] who analyzed the parameterized complexity of computing envy-free and Pareto-efficient allocations focusing on the parameters number n of agents, number m of items, and the maximum utility value z. While other works for different parameters and variants of envy-freeness and efficiency have been performed [13, 14, 21], only Bliem [7] provided some insight into the kernelization of the problem, showing a linear problem kernel with respect to m for additive preferences and (presumably) non-existence of polynomial kernels for so-called dichotomous preferences. There is a significant amount of kernelization results for less closely related collective decision making problems (with kernels mostly appearing as side results) [4, 7, 12, 17, 30, 34]. We note that *systematic* studies of problem kernelization, where a set of parameters is systematically analyzed with respect to existance and quality of problem kernels, as done for graph modification problems, are still rare in context of fair division (or generally in the field Computational Social Choice; see, e.g., Key Question 4 of Bredereck et al. [11]), thus we view our kernelization results as additional contributions on their own.

1.2 Our Contributions

Our results are summarized in Table 1. Of particular interest is the parameter solution size k: in particular, while for general additive preferences SAS is W[2]-hard and in XP, for approval-based preferences and for Borda preferences, however, SAS is in FPT but (presumably) admits no polynomial problem kernel; for t-approval and t-veto preferences SAS even admits polynomial problem kernels. For the number m of items, SAS is (trivially) fixed-parameter tractable. We show, however, that SAS (presumably) admits no polynomial problem kernel for the number m of items in case of general preferences or Borda preferences, while it does so for t-approval (t-veto) preferences. For general preferences SAS is W[1]-hard even for the combined parameter $n + k$, while it is fixed-parameter tractable for n alone for approval preferences. We also consider the parameter $m - k$, which is dual to the solution size, and observe an interesting difference between t-approval and t-veto preferences: SAS remains W[1]-hard for t-approval preferences but becomes fixed-parameter tractable for t-veto preferences. This shows that smallest agreeable subsets for dense approval preferences are easier to find. Indeed, t-approval preferences and t-veto preferences appear "symmetric" at first glance: A small constant number of disapprovals per agent, however, is easier to handle compared to a small constant number of approvals per agent.[1]

Remark 2. We wish to stress that, in our view, we study only the very natural parameters of the input. In particular, instances with a small set of items to be selected (our parameter k) correspond to limited budgets or capacities in relevant usecases; while instances with small number of agents are also well-motivated in the context of resource allocation.

2 Preliminaries

We use basics from parameterized algorithmics [15]. We have the hierarchy FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \cdots \subseteq$ XP, where problems in XP are solvable in polynomial-time for a constant-valued parameter. We call problems in FPT tractable since the degree of the polynomial does not depend on the constant value of the parameter. Problems which are hard for W[1], W[2], etc., are called intractable as they presumably do not admit running times as problems in FPT. A problem kernel is an algorithm that maps in polynomial time every input instance to an equivalent instance of size upper bounded by the input instance's parameter. It is well-known that a decidable problem is in FPT if and only if it admits a problem kernel. Thus, the class FPT is often partitioned into those problems which admit polynomial-sized problem kernels, and those which do presumably not (unless NP $\subseteq$ coNP/poly). Problem kernels often rely on data reduction rules. For SAS, for instance, we have the following (recall that an agreeable subset is agreeable for *all* agents).

Reduction Rule 1. *If there are agents with the same utility function/preferences, then delete all but one of them.*

Details to results marked with $\star$ are deferred to a full version.

3 General Additive Utilities

We denote SAS with general additive utility-based preferences by CARDINAL-SAS. Manurangsi and Suksompong [28] proved that CARDINAL-SAS with binarily-encoded utilities is NP-hard even for two agents. We prove that, even for unary encoding, intractability for the parameters n, k, and $n + k$ remains.

Theorem 1. *For unary encoding,* CARDINAL-SAS *is NP-hard and (i) in XP and W[2]-hard when parameterized by the solution size k, (ii) in XP when parameterized by the number n of agents, and (iii) W[1]-hard when parameterized by $n + k$.*

We prove Theorem 1(i) to (iii) through Propositions 1 to 3. We start with the solution size k as the parameter.

Proposition 1. *For unary encoding,* CARDINAL-SAS *is NP-hard, W[2]-hard when parameterized by the solution size k, and solvable in $O(\binom{m}{k} \cdot (n \cdot m))$ time.*

We will reduce from the following.

Problem 2. HITTING SET (HS)

Input: A universe U, a family $\mathcal{F}$ of subsets of U of maximum size $d \in \mathbb{N}$, and $k \in \mathbb{N}$.
Question: Is there a subset $U' \subseteq U$ with $|U'| \leq k$ such that $F \cap U' \neq \emptyset$ for every $F \in \mathcal{F}$?

HITTING SET is NP-hard and W[2]-complete when parameterized by the solution size k [20].

Proof. A trivial brute-force algorithm that tests every k-sized subset of C runs in $O(\binom{m}{k} \cdot (n \cdot m))$ time.

We provide a parameterized reduction from HS when parameterized by the solution size k. Note that the reduction is also

[1] For illustration, note that for cases with much more items compared to agents, for t-veto preferences we will have many items that are approved by everyone while for t-approval preferences there will be many items disapproved by everyone. Having an item disapproved by everyone (or not) does not change the set of smallest agreeable subsets at all. Items approved by everyone are, however, very helpful. More than $(m - t)/2$ such items even guarantee a smallest agreeable subset, which only consists of items approved by everyone.

Table 1. Summary of our results. A "–" means that the corresponding combination is not applicable. "PPK" stands for a polynomial problem kernel, while NoPPK means that, presumably, no PPK exists. ($*$: Thm. 1) (♠: Prop. 1) (♣: Prop. 2) (♢: Prop. 3) (♡: Cor. 2) ($\circ$: Thm. 2) (†: Prop. 5) (‡: Cor. 1) (¶: Prop. 4) ($\times$: Cor. 3) ($\diamond$: Obs. 2) ($\triangle$: Obs. 1) (■: Thm. 3) (❦: Obs. 3) (#: Obs. 4) ($: Thm. 4) (%: Cor. 5)

-SAS	Comput. Complex.	Parameterized Complexity k	n	$n+k$	m	$m-k$
CARDINAL	NP-c. ($*$)	XP (♠), W[2]-h. (♠)	XP (♣), W[1]-h. (♢)	XP, W[1]-h. (♢)	FPT, NoPPK (♡)	XP (♠), W[1]-h. (♡)
APPROVAL	NP-c. ($\circ$)	FPT (†), NoPPK (♡)	FPT (‡), NoPPK (♡)	FPT, $O(kn)$ PPK (¶)	FPT, NoPPK (♡)	XP (♠), W[1]-h. (♡)
1-APPROVAL	P.-time ($\circ$)	–				
t-APPROVAL	NP-c. ($\circ$)	FPT, PPK ($\times$)	FPT, $O(t \cdot n)$ PPK ($\diamond$)		FPT, $O(m^t)$ PPK ($\triangle$)	XP (♠), W[1]-h. (♡)
t-VETO	NP-c. ($\circ$)	FPT, PPK ($\times$)	FPT, $O(n^2 t)$ PPK (■)	FPT, $O(tn+k)$ PPK (❦)	FPT, $O(m^t)$ PPK (#)	FPT, PPK (#)
1-VETO	P.-time ($\circ$)	–				
BORDA	NP-c. ($)	FPT (%), NoPPK ($)	XP (♣), *FPT open*	FPT, $O(kn)$ PPK (%)	FPT (%), NoPPK ($)	FPT (%), NoPPK ($)

a polynomial-time many-one reduction for the unparameterized problems. Given an instance $(U, \mathcal{F}, k)$ of HS, we construct an instance $(C, A, (u_a)_{a \in A}, k')$ of CARDINAL-SAS, as follows.

For each element $x \in U$, we add an item c_x. Moreover, we add one special item c^*. For each set $F \in \mathcal{F}$, we add an agent a_F with utility

$$u_{a_F}(c) = \begin{cases} 1, & \text{if } c = c_x \wedge x \in F, \\ |F| - 1, & \text{if } c = c^*, \\ 0, & \text{otherwise.} \end{cases}$$

Add an agent a^* with utility $u_{a^*}(c^*) = 1$ and $u_{a^*}(c) = 0$ for all $c \neq c^*$. Set $k' = k + 1$. We claim that $(U, \mathcal{F}, k)$ is a *yes*-instance of HS if and only if $(C, A, (u_a)_{a \in A}, k')$ is a *yes*-instance of CARDINAL-SAS (see full version). □

Recall that CARDINAL-SAS with binarily-encoded utilities is NP-hard even for two agents [28, Theorem 5]. The upcoming Proposition 2 shows that CARDINAL-SAS with unary encoding is in XP when parameterized by the number n of agents.

Proposition 2 ([28, Theorem 5])**.** *For unary encoding,* CARDINAL-SAS *is solvable in* $O(m \cdot k \cdot (\sigma + 1)^n)$ *time, where* $\sigma = \max_{a \in A} \sum_{c \in C} u_a(c)$ *is the maximum sum of utilities over all agents.*

Remark 3. There is also a straightforward ILP formulation for CARDINAL-SAS with the following constraints:

$$\begin{aligned} &\sum_{c_i \in C} u_a(c_i) \cdot (2x_i - 1) \geq 0 \quad && \forall a \in A \\ &\sum_{c_i \in C} x_i \leq k && \\ &x_i \in \{0, 1\} && \forall c_i \in C \end{aligned} \tag{1}$$

Due to Eisenbrand and Weismantel [22], we know that this is solvable by $((n + 1) \cdot \Gamma)^{O(n^2)} \cdot \log_2(n \cdot \Gamma) \cdot m$, where $\Gamma = \max_{a \in A} \max_{c \in C} u_a(c)$, and hence in XP-time. While the dynamic program yields a better theoretical running time, the ILP might nevertheless work much better in practice. (For the dynamic programming approach, one may naturally expect the worst case running time for arbitrary instances, while ILP is typically much faster.) ◁

We show next that improving this to fixed-parameter tractability is unlikely. In fact, we prove a stronger statement.

Proposition 3. *For unary encoding,* CARDINAL-SAS *is W[1]-hard when parameterized by* $n + k$*.*

The following proof is inspired by the work of Fluschnik et al. [24]. The reduction is from the following W[1]-complete [23, 32] problem.

Problem 3. MULTICOLORED CLIQUE (MCC)

Input: An undirected, k-partite graph $G = (V^1 \uplus \cdots \uplus V^k, E)$ with $k \in \mathbb{N}$.

Question: Does G contain a clique $X = (W, F)$ with $|W \cap V^i| = 1$ for every $i \in \{1, \ldots, k\}$?

Proof. Let $G = (V^1 \uplus \cdots \uplus V^k, E)$ be an instance of MCC. We assume that $V^i = \{v_1^i, \ldots, v_\ell^i\}$ for all $i \in \{1, \ldots, k\}$. For every $i, j \in \{1, \ldots, k\}$, $i \neq j$, denote by $E^{i,j}$ the set of edges between V^i and V^j and $m_{i,j} := |E^{i,j}|$.

For each vertex and edge, we create an item (and slightly abuse notation). Let C_0 denote this set. We introduce one special item z, and let $C := C_0 \cup \{z\}$. We create the following agents:

(i) Agents $a_1, \ldots, a_k$ where

$$u_{a_i}(x) := \begin{cases} 1, & \text{if } x \in V^i, \\ 0, & \text{if } x \in C_0 \setminus V^i, \\ \ell - 1, & \text{if } x = z. \end{cases}$$

(ii) Agents $a_{i,j}$, $\{i, j\} \in \binom{\{1,\ldots,k\}}{2}$, where

$$u_{a_{i,j}}(x) := \begin{cases} 1, & \text{if } x \in E^{i,j}, \\ 0, & \text{if } x \in C_0 \setminus E^{i,j}, \\ m_{i,j} - 1, & \text{if } x = z. \end{cases}$$

(iii) Agents $a^1_{(i,j)}$ and $a^2_{(i,j)}$, $(i, j) \in \{1, \ldots, k\} \times \{1, \ldots, k\}$, $i \neq j$, where $u_{a^1_{(i,j)}}(x)$ is defined as

$$\begin{cases} p, & \text{if } x = v_p^i \in V^i, \\ \ell - p + 1, & \text{if } x = \{v_p^i, v_q^j\} \in E^{i,j}, \\ 0, & \text{if } x \in C_0 \setminus (V^i \cup E^{i,j}), \\ -2(\ell + 1) + \sum\limits_{x \in C_0} u_{a^1_{(i,j)}}(x), & \text{if } x = z, \end{cases}$$

and $u_{a^2_{(i,j)}}(x)$ is defined as

$$\begin{cases} \ell - p + 1, & \text{if } x = v_p^i \in V^i, \\ p, & \text{if } x = \{v_p^i, v_q^j\} \in E^{i,j}, \\ 0, & \text{if } x \in C_0 \setminus (V^i \cup E^{i,j}), \\ -2(\ell + 1) + \sum\limits_{x \in C_0} u_{a^2_{(i,j)}}(x), & \text{if } x = z. \end{cases}$$

(iv) Add agent a^* only approving z, i.e., $u_{a^*}(x) = 1$ if $x = z$, and 0 otherwise.

Set $k' = k + \binom{k}{2} + 1$. Note that the number of agents is $k + \binom{k}{2} + 2(k-1)k + 1$. It remains to show the correctness of the reduction (which we defer to a full version). □

Bounded Utilities. Let $\Gamma = \max_{a \in A} \max_{c \in C} u_a(c)$ denote the maximum utility that some item receives from some agent. Combining the parameters n and k with Γ leads to tractability results. When combined with parameter n fixed-parameter tractability follows from Remark 3.

Corollary 1. CARDINAL-SAS *is fixed-parameter tractable when parameterized by* $n + \Gamma$.

For the solution size k, observe that any unsatisfied agent can have at most $2k\Gamma$ items of strictly positive utility, since otherwise k items even with utility Γ each cannot satisfy the agent. In fact, we even have the following.

Reduction Rule 2. *If there is an agent with sum of utilities larger than* $2k\Gamma$*, then return a trivial* no-*instance.*

Now, there cannot be too many items with non-zero utility and items that get zero utility from every agent are irrelevant.

Reduction Rule 3. *If there is an item* c *that gets zero utility from every agent, then delete* c*.*

After the exhaustive application of Reduction Rule 2 and Reduction Rule 3, we have an instance where every agent's utilities sum up to at most $2k\Gamma$ and there is no item of zero utility for all the n agents, and hence a problem kernel that is polynomial in $n + k + \Gamma$.

Proposition 4. CARDINAL-SAS *admits a problem kernel of size* $O(k\Gamma n)$*.*

For any not-yet-satisfied agent, we can branch over the at-most $2k\Gamma$ items with strictly positive utility.

Proposition 5. CARDINAL-SAS *is fixed-parameter tractable for the solution size* k *combined with the maximum utility* Γ *and solvable in* $O((2k\Gamma)^k \cdot k\Gamma n)$ *time.*

Proof. Let $(A, C, (u_a)_{a \in A}, k)$ be an instance of CARDINAL-SAS with maximum utility Γ. Due to Reduction Rule 2, we can assume that each agent has at most $2k\Gamma$ items with positive utility. Moreover, due to Reduction Rule 3, we can assume that we have at most $2k\Gamma \cdot n$ many items.

We proceed in at most k branchings where we maintain a partial solution S. Initially let $S = \emptyset$. In each branching we consider those agents which are currently not satisfied with S; if all agents are satisfied, then we output *yes*. Otherwise, we select one of them arbitrarily and branch into selecting one of its at most $2k\Gamma$ items with positive utility which is not in S. If after k branchings we still have some unsatisfied agents then we output *no*. In each branching, we branch over at most $2k\Gamma$ items, and we stop after k branchings, yielding a running time of $O((2k\Gamma)^k \cdot k\Gamma n)$. □

Note that binary utilities are a special case of general utilities where the maximum value is bounded by 1. This is the approval setting, which we will discuss in the next section.

4 Approval Scores

We continue with the binary setting, where $u\colon C \to \{0, 1\}$, which is equivalent to the setting where agents specify approvals. We hence express utilities by $B_u\colon A \to 2^C$, where $u_a(c) = 1 \iff c \in B_u(a)$ for every agent $a \in A$ and candidate $c \in C$. We first study the general approval setting (Section 4.1), then we fix the number of approvals and disapprovals (Sections 4.2 and 4.3).

4.1 Approval Voting

The APPROVAL-SAS problem is quite tractable: While it is NP-hard in general, it is fixed-parameter tractable for each of n and k (see Propositions 4 and 5). However, polynomial problem kernelization is presumably not possible regarding m and n. We start with the NP-hardness result, improving the known general hardness [28]:

Theorem 2. APPROVAL-SAS *is NP-hard if each agent approves* $d \in \{2, \ldots, m-2\}$ *items, and linear-time solvable if each agent approves one item or if each each agent approves (at least) all but one item.*

Proposition 6 ($\star$)**.** *Both* 1-APPROVAL-SAS *and* 1-VETO-SAS *are linear-time solvable.*

The NP-hardness in Theorem 2 is due to a reduction from HS.

Lemma 1 ($\star$)**.** *There is a polynomial-time algorithm that maps every instance* $(U, \mathcal{F}, k)$ *of* HS *with maximum set size* d *to an equivalent instance* (C, A, u, k') *of* APPROVAL-SAS *with* $|A| = |\mathcal{F}|+1$, $|C| = |U|+2(d-2)$*, and* $k' = k+d-2$*, where each agents approves* $2d-2$ *items.*

We know that HS is NP-complete for $d \geq 2$, and hence Theorem 2 with $d \geq 2$ follows. Dom et al. [18] proved that, unless NP $\subseteq$ coNP/poly, HS admits no problem kernel of size polynomial in $|U|$ (note that $k, d \leq |U|$) or $|\mathcal{F}|$ (see also [15, Chapter 15]). Moreover, HS already with $d = 2$ is W[1]-hard when parameterized by the number of elements *minus* the solution size (due to the connection to INDEPENDENT SET). Thus, we additionally have the following.

Corollary 2. APPROVAL-SAS *is W[1]-hard when parameterized by* $m - k$ *even for two approvals. Unless NP* $\subseteq$ *coNP/poly,* APPROVAL-SAS *admits no problem kernel of size polynomial in* m *or* n*.*

It remains to show that APPROVAL-SAS is NP-hard when every agent approves all but two items, which we call 2-VETO-SAS. To this end, consider the following variant of INDEPENDENT SET, which we call EXACT INPEDENDENT SET (EIS), where given an undirected graph $G = (V, E)$ with an even number of vertices and $k = |V|/2$, the question is whether there is a subset $V' \subseteq V$ with $|V'| \geq k$ such that no two vertices in V' are adjacent. EIS is clearly NP-complete. Reduce this variant to 2-VETO-SAS to obtain the following.

Proposition 7 ($\star$)**.** 2-VETO-SAS *is NP-hard.*

4.2 t-Approval

For APPROVAL-SAS, we showed that there are no polynomial problem kernels regarding the number n of agents or the number m of items unless NP $\subseteq$ coNP/poly. Fixing the number t of approvals, however, leads to problem kernels of size polynomial in each of n and k. A polynomial problem kernel regarding m is immediate: If Reduction

Rule 1 is inapplicable, then every two agents approve different items. Hence, since there are at most $\binom{m}{t}$ possibilities to approve t items from a set of m items, we have the following.

Observation 1. t-APPROVAL-SAS *admits a problem kernel of size* $O(\binom{m}{t})$.

A polynomial problem kernel regarding n is also not hard: If Reduction Rule 3 is inapplicable, then the set of items is the union of the t approved items from each agents. Thus, we have the following.

Observation 2. t-APPROVAL-SAS *admits a problem kernel of size* $O(t \cdot n)$.

We obtain a problem kernel via the t-HITTING SET (t-HS) problem, where each set size is bounded by t.

Lemma 2 (⋆)**.** *There is a polynomial-time algorithm that maps every instance* (C, A, u, k) *of* t-APPROVAL-SAS *to an equivalent instance* $(U, \mathcal{F}, k)$ *of* t-HS.

t-HS admits a problem kernel of size $O(k^t)$ [1, 6]. Thus, due to Lemma 2:

Corollary 3. t-APPROVAL-SAS *admits a problem kernel of size polynomial in* k.

4.3 t-Veto Scores

We have already seen that 2-VETO-SAS is NP-hard. For t-VETO-SAS, we have the following important observation.

Observation 3 (⋆)**.** *If* $k < (m-t)/2$ *or* $k \geq (m-t)/2 + t$*, then we have a trivial* no- *or* yes-*instance, respectively.*

As a consequence, we have that

$$(m-t)/2 \leq k < (m+t)/2 \text{ and, consequently,} \quad k, m-k \in \Theta(m). \quad (2)$$

As before, since there are at most $\binom{m}{t}$ possibilities to disapprove t items from a set of m items, together with Reduction Rule 1, we hence have the following.

Observation 4. t-VETO-SAS *admits a problem kernel of size* $O(\binom{m}{t} \cdot t)$.

In contrast to t-APPROVAL-SAS, t-VETO-SAS is fixed-parameter tractable for the parameter $m-k$ which can be seen by a simple branching algorithm.

Proposition 8 (⋆)**.** t-VETO-SAS *is solvable in* $O(t^{m-k} \cdot (n \cdot m))$ *time.*

Interestingly, t-VETO-SAS is NP-hard even if $m/2 - k$ is constant (Proposition 7) but polynomial-time solvable if $m-k$ is constant (Proposition 8). We leave open the complexity of t-VETO-SAS with parameter $\varepsilon \cdot m/2 - k$ for $1 < \varepsilon < 2$.

Finally, we also have a polynomial problem kernel regarding n for t-VETO-SAS. However, it is quadratic instead of linear as for t-APPROVAL-SAS.

Theorem 3. t-VETO-SAS *admits a problem kernel of size* $O(n^2 \cdot t)$.

Proof. Let Y denote the set of items that are everywhere approved, and let Z denote the set of items that are at least once not approved. We know that $|Z| \leq n \cdot t$. Thus, if $|Y| \geq (n-1) \cdot t$, then we can only pick from Y. This is due to the fact that all other items any agent a approves next to Y must be from Z, which are without a's vetoed items at most $(n-1) \cdot t$ many. Thus, we can pick $\min k, |Y|$ many items from Y and check whether they form an agreeable subset. Otherwise, we have $|Y| < (n-1) \cdot t$, and thus $m = |Z| + |Y| < n \cdot t + (n-1) \cdot t$. □

5 Borda Scores

Our last preference model assumes that each agent has a linear order over the items and the item at position i has utility value $m - i$. Formally, we have $u_a \colon C \to \{0, 1, \ldots, |C| - 1\}$ where $\bigcup_{c \in C} u_a(c) = \{0, 1, \ldots, |C| - 1\}$ for every agent $a \in A$. We also identify the utilities with rankings, where an agent a ranks c on position i if and only if $u_a(c) = m - i$. While such preferences are very natural [31], they have not been considered in context of SAS.

What follows is not covered neither by Theorem 2 nor by the work of Manurangsi and Suksompong [28]. In fact, similar to coalitional manipulation for Borda voting [5, 16] (yet requiring a completely different and new reduction), it was technically most challenging to obtain hardness for the Borda case.

Theorem 4. BORDA-SAS *is NP-complete and, unless NP* $\subseteq$ *coNP/poly, admits no problem kernel of size polynomial in* m.

We introduce the following problem from which we will reduce to BORDA-SAS:

Problem 4. WEAK-MAJORITY-SAT (WM-SAT)
Input: A boolean CNF formula ϕ over a set X of N variables.
Question: Is there a truth assignment such that in each clause at least half of the literals evaluate to true?

We have the following polynomial parameter transformation [8] from SAT regarding the number of variables.

Lemma 3. *There is a polynomial-time algorithm that maps every instance* (ϕ, X) *of* SAT *to an equivalent instance* (ϕ', X') *of* WM-SAT *such that* $|X'| \leq 2|X|$.

Unless NP ⊆ coNP/poly, SAT admits no problem kernel of size polynomial in the number of variables [25, 27], implying the following.

Corollary 4. WM-SAT *is NP-hard and, unless NP* $\subseteq$ *coNP/poly, admits no problem kernel of size polynomial in* N.

We further assume that no variable appears both negated and unnegated in a clause (as to the reduction from SAT).

Construction 1. Let $I = (X, \phi)$ be an instance of WM-SAT with $|X| = N$ and M clauses. We construct an instance $I' := (C, A, (u_a)_{a \in A}, k)$ of BORDA-SAS as follows. Let $C := X \cup Y$ where $X := \{x_i, \overline{x_i} \mid i \in \{1, \ldots, N\}\}$ and $Y := \{y_i, \overline{y_i} \mid i \in \{1, \ldots, N\}\}$. Note that $m := |C| = 4N$. Let the agents $A' := \{a_{x,y}, a_{y,x}\} \cup \{a_x^i, a_y^i \mid i \in \{1, \ldots, N\}\} \cup \{a^{i,j} \mid i \neq j \in \{1, \ldots, N\}\}$ and $D := \{d_i \mid i \in \{1, \ldots, M\}\}$. See Fig. 1 for the preferences. Let $\widehat{A'}$ contain an agent $\widehat{a}$ for each agent $a \in A'$ with the mirrored profile, i.e., if a ranks c on position i, then $\widehat{a}$ ranks c on position $m + 1 - i$. Let $A := A' \cup \widehat{A'} \cup D$. Let $k := 2N$. ⋄

In what follows, we discuss the case of I' being a *yes*-instance. We start with the following crucial fact that is due to the mirrored agents.

$a_{x,y}$:	$x_1, \overline{x_1}$	$x_2, \overline{x_2}$	$\cdots\cdots$	$x_N, \overline{x_N}$	$\overline{y_1}, y_1$	$\overline{y_2}, y_2$	$\cdots\cdots$	$\overline{y_N}, y_N$			
$a_{y,x}$:	$\overline{y_1}, y_1$	$\overline{y_2}, y_2$	$\cdots\cdots$	$\overline{y_N}, y_N$	$x_1, \overline{x_1}$	$x_2, \overline{x_2}$	$\cdots\cdots$	$x_N, \overline{x_N}$			
a_x^i:	$R_{\mathrm{x}}(i)$	$R_{\mathrm{x}}(1)$	$\cdots\cdots$	$R_{\mathrm{x}}(i-1)$	$R_{\mathrm{x}}(i+1)$	$\cdots\cdots$	$R_{\mathrm{x}}(N)$				
a_y^i:	$R_{\mathrm{y}}(i)$	$R_{\mathrm{x}}(1)$	$\cdots\cdots$	$R_{\mathrm{x}}(i-1)$	$R_{\mathrm{x}}(i+1)$	$\cdots\cdots$	$R_{\mathrm{x}}(N)$				
$a^{i,j}$:	$R_{\mathrm{x}}(i)$	$R_{\mathrm{x}}(1)$	$\cdots$	$R_{\mathrm{x}}(i-1)$	$R_{\mathrm{x}}(i+1)$	$\cdots$	$R_{\mathrm{x}}(j-1)$	$R_{\mathrm{x}}(j+1)$	$\cdots$	$R_{\mathrm{x}}(N)$	$R_{\mathrm{x}}(j)$
d^i:	$\ell^i_{x,1}, \dots, \ell^i_{x,\lvert K_i\rvert}$	$\ell^i_{y,1}, \dots, \ell^i_{y,\lvert K_i\rvert}$	$R_{\mathrm{x}}(i_1)$	$\cdots$	$R_{\mathrm{x}}(i_{N-\lvert K_i\rvert})$	$\overline{\ell^i_{x,1}, \dots, \ell^i_{x,\lvert K_i\rvert}}$	$\overline{\ell^i_{y,1}, \dots, \ell^i_{y,\lvert K_i\rvert}}$				

Figure 1. Illustration to Construction 1, where the candidates are ordered from left to right. Here, $R_{\mathrm{x}}(i) \coloneqq x_i \prec \overline{x_i} \prec \overline{y_i} \prec y_i$ and $R_{\mathrm{y}}(i) \coloneqq y_i \prec \overline{y_i} \prec \overline{x_i} \prec x_i$. In this example, we have $a^{i,j}$ with $1 < i < j < N$.

Lemma 4. *If I' is a* yes-*instance, then for every solution $C' \subseteq C$ it holds true that (i) for every agent $a \in A'$ we have that $u_a(C') = u_a(C \setminus C')$ and (ii) $|C'| = |C \setminus C'|$.*

Proof. For every $a \in A'$, for a's mirror $\widehat{a}$ we have

$$u_{\widehat{a}}(C') = \sum_{c \in C'} u_{\widehat{a}}(c) = \sum_{c \in C'} (m - 1 - u_a(c)) = |C'| \cdot (m-1) - \sum_{c \in C'} u_a(c). \quad (3)$$

Together with the two facts that $u(C') \geq u(C \setminus C')$, i.e., $\sum_{c \in C'} u_a(c) \geq \sum_{c \in C \setminus C'} u_a(c)$, and $|C'| \leq 2N \leq |C \setminus C'|$, we get the following:

$$\begin{aligned}(3) &\leq |C'| \cdot (m-1) - \sum_{c \in C \setminus C'} u_a(c) && (4)\\ &\leq |C \setminus C'| \cdot (m-1) - \sum_{c \in C \setminus C'} u_a(c) && (5)\\ &= \sum_{c \in C \setminus C'} (m - 1 - u_a(c)) = \sum_{c \in C \setminus C'} u_{\widehat{a}}(c) = u_{\widehat{a}}(C \setminus C').\end{aligned}$$

Thus, if (i) or (ii) is false, then we get "$<$" in (4) or (5), resp., contradicting the fact that $u_{\widehat{a}}(C') \geq u_{\widehat{a}}(C \setminus C')$. □

The next property is due to agents $a_{x,y}$ and $a_{y,x}$ combined with Lemma 4.

Lemma 5. *If I' is a* yes-*instance, then for every solution $C' \subseteq C$ it holds true that $|C' \cap X| = |C' \cap Y|$.*

Proof. Suppose not, that is, $|C' \cap X| \neq |C' \cap Y|$. Recall that $|C'| = k = |C|/2$ due to Lemma 4. Thus, we know that $|C' \cap X| = |(C \setminus C') \cap Y|$ and $|C' \cap Y| = |(C \setminus C') \cap X|$. Consider $a_{x,y}$. Let $(p_i, q_i)_{i=1}^{|C' \cap X|}$ be such that $\bigcup_{i=1}^{|C' \cap X|} x_{p_i} = C' \cap X$ and $\bigcup_{i=1}^{|C' \cap X|} y_{q_i} = (C \setminus C') \cap Y$. Analogously, let $(p_i', q_i')_{i=1}^{|C' \cap Y|}$ be such that $\bigcup_{i=1}^{|C' \cap Y|} x_{p_i'} = (C \setminus C') \cap X$ and $\bigcup_{i=1}^{|C' \cap Y|} y_{q_i'} = C' \cap Y$. We know that (due to Lemma 4)

$$\begin{aligned}0 &= u_{a_{x,y}}(C') - u_{a_{x,y}}(C \setminus C')\\ &= \sum_{i=1}^{|C' \cap X|} \left(u_{a_{x,y}}(x_{p_i}) - u_{a_{x,y}}(y_{q_i})\right)\\ &\quad + \sum_{i=1}^{|C' \cap Y|} \left(u_{a_{x,y}}(y_{q_i'}) - u_{a_{x,y}}(x_{p_i'})\right)\end{aligned} \quad (6)$$

Moreover, we have that

$$\begin{aligned}\forall x \in X: &\quad u_{a_{x,y}}(x) - m/2 = u_{a_{y,x}}(x), \text{ and}\\ \forall y \in Y: &\quad u_{a_{x,y}}(y) + m/2 = u_{a_{y,x}}(y).\end{aligned} \quad (7)$$

Thus (due to Lemma 4):

$$\begin{aligned}0 &\overset{(6)}{=} \sum_{i=1}^{|C' \cap X|} \left(u_{a_{y,x}}(x_{p_i}) - u_{a_{y,x}}(y_{q_i})\right)\\ &\quad + \sum_{i=1}^{|C' \cap Y|} \left(u_{a_{y,x}}(y_{q_i'}) - u_{a_{y,x}}(x_{p_i'})\right)\\ &\overset{(7)}{=} \sum_{i=1}^{|C' \cap X|} \left(u_{a_{x,y}}(x_{p_i}) - u_{a_{x,y}}(y_{q_i}) - m\right)\\ &\quad + \sum_{i=1}^{|C' \cap Y|} \left(u_{a_{x,y}}(y_{q_i'}) - u_{a_{x,y}}(x_{p_i'}) + m\right)\end{aligned} \quad (8)$$

Recall that since $a_{x,y} \in A'$, we have due to Lemma 4 that $\sum_{i=1}^{|C' \cap X|} \left(u_{a_{x,y}}(x_{p_i}) - u_{a_{x,y}}(y_{q_i})\right) = \sum_{i=1}^{|C' \cap Y|} \left(u_{a_{x,y}}(x_{p_i'}) - u_{a_{x,y}}(y_{q_i'})\right)$. Thus, it remains to compare $|C' \cap X| \cdot m$ with $|C' \cap Y| \cdot m$, which is different due to our assumption that $|C' \cap X| \neq |C' \cap Y|$. Hence, we get

$$\begin{aligned}(8) &= \sum_{i=1}^{|C' \cap X|} \left(u_{a_{x,y}}(x_{p_i}) - u_{a_{x,y}}(y_{q_i})\right) - |C' \cap X| \cdot m\\ &\quad + \sum_{i=1}^{|C' \cap Y|} \left(u_{a_{x,y}}(y_{q_i'}) - u_{a_{x,y}}(x_{p_i'})\right) + |C' \cap Y| \cdot m\\ &\neq \sum_{i=1}^{|C' \cap X|} \left(u_{a_{x,y}}(x_{p_i}) - u_{a_{x,y}}(y_{q_i})\right)\\ &\quad + \sum_{i=1}^{|C' \cap Y|} \left(u_{a_{x,y}}(y_{q_i'}) - u_{a_{x,y}}(x_{p_i'})\right) = 0,\end{aligned}$$

a contradiction. □

The next property is due to agents a_x^i, a_y^i and $a^{i,j}$, combined with Lemmas 4 and 5.

Lemma 6. *If I' is a* yes-*instance, then for every solution $C' \subseteq C$ it holds true that for every $i \in \{1, \dots, N\}$ we have that $|C' \cap \{x_i, \overline{x_i}\}| = |C' \cap \{y_i, \overline{y_i}\}| = 1$.*

Proof. Suppose not, i.e., there is $i \in \{1,\ldots,N\}$ we have that $|C' \cap \{x_i, \overline{x_i}\}| \in \{0,2\}$. Note that due to a^i_x, a^i_y and the fact that $u_{a^i_x}(C') - u_{a^i_x}(C \setminus C') = u_{a^i_y}(C') - u_{a^i_y}(C \setminus C') = 0$, we have that if $x_i, \overline{x_i} \in C'$, then also $y_i, \overline{y_i} \in C'$. Since $|C' \cap X| = |C' \cap Y| = N$ (due to Lemma 5), there is $j \in \{1,\ldots,N\}$ such that $C' \cap \{x_j, \overline{x_j}\} = \emptyset$. Then, due to Lemma 4, $0 = u_{a^{i,j}}(C') - u_{a^{i,j}}(C \setminus C') \neq u_{a^{j,i}}(C') - u_{a^{j,i}}(C \setminus C') = 0$, a contradiction.
The proof works analogously for $y_i, \overline{y_i}$. □

The next lemma states that if x_i is contained in the agreeable subset, then also y_i is, and vice versa.

Lemma 7. *If I' is a* yes-*instance, then for every solution $C' \subseteq C$ it holds true that for every $i \in \{1,\ldots,N\}$ we have that $x_i \in C' \iff y_i \in C'$.*

Proof. Suppose that for some $i \in \{1,\ldots,N\}$, $x_i \in C'$ (implying that $\overline{x_i} \in C \setminus C'$) and $\overline{y_i} \in C'$ (implying that $y_i \in C \setminus C'$). Then, due to Lemma 4, $0 = u_{a^i_x}(C') - u_{a^i_x}(C \setminus C') \neq u_{a^i_y}(C') - u_{a^i_y}(C \setminus C') = 0$, a contradiction. □

The following is the last lemma before we prove Theorem 4. It states that every agreeable subset contains at least half of the items corresponding to each clause's literals.

Lemma 8. *If I' is a* yes-*instance, then for every solution $C' \subseteq C$ it holds true that for every $i \in \{1,\ldots,M\}$ we have that $|C' \cap \{\ell^i_{x,1},\cdots,\ell^i_{x,|K_i|}\}| \geq |K_i|/2$.*

Proof. Clearly, for every $i \in \{1,\ldots,M\}$, if $|C' \cap \{\ell^i_{x,1},\cdots,\ell^i_{x,|K_i|}\}| < |K_i|/2$, then also $|C' \cap \{\ell^i_{y,1},\cdots,\ell^i_{y,|K_i|}\}| < |K_i|/2$. Let $\Lambda_i := \{j \in \{1,\ldots,|K_i|\} \mid \ell^i_{x,j} \in C'\}$ and $\overline{\Lambda_i} := \{1,\ldots,|K_i|\} \setminus \Lambda_i$. Note that we have that

$$
\begin{aligned}
&\frac{1}{2}\left(u_{d^i}(C') - u_{d^i}(C \setminus C')\right) \\
&= \sum_{j \in \Lambda_i} \ell^i_{x,j} + \sum_{j \in \overline{\Lambda_i}} \overline{\ell^i_{x,j}} - \left(\sum_{j \in \Lambda_i} \overline{\ell^i_{x,j}} + \sum_{j \in \overline{\Lambda_i}} \ell^i_{x,j} \right) \\
&= \sum_{j \in \Lambda_i} \left(\ell^i_{x,j} - \overline{\ell^i_{x,j}}\right) + \sum_{j \in \overline{\Lambda_i}} \left(\overline{\ell^i_{x,j}} - \ell^i_{x,j}\right) \\
&= \sum_{j \in \Lambda_i} \left(\ell^i_{x,j} - \overline{\ell^i_{x,j}}\right) - \sum_{j \in \overline{\Lambda_i}} \left(\ell^i_{x,j} - \overline{\ell^i_{x,j}}\right) \\
&= \kappa \cdot |\Lambda_i| - \kappa \cdot |\overline{\Lambda_i}|,
\end{aligned}
$$

where $\kappa = 4(N - |K_i|) + 2|K_i|$, which is the difference between the Borda-scores of $\ell^i_{x,j}$ and $\overline{\ell^i_{x,j}}$ for every $j \in \{1,\ldots,K_i\}$. Hence, if $|C' \cap \{\ell^i_{x,1},\cdots,\ell^i_{x,|K_i|}\}| < |K_i|/2$, then we have that $u_{d^i}(C') < u_{d^i}(C \setminus C')$. □

We are set to prove Theorem 4.

Proof of Theorem 4. Given an instance $I = (X, \phi)$ of WM-SAT with $|X| = N$ and M clauses, construct instance $I' := (A, C, (u_a)_{a \in A}, k)$ of BORDA-SAS using Construction 1. We prove that I is *yes*-instance if and only if I' is a *yes*-instance.

($\Rightarrow$) Let $f \colon X \to \{\top, \bot\}$ be an assignment that for each clause evaluates at least half of its literals to $\top$. Let $C' \subseteq C$ be the committee with for every $i \in \{1,\ldots,N\}$, we have $x_i, y_i \in C'$ if $f(x_i) = \top$ and $\overline{x_i}, \overline{y_i} \in C'$ if $f(x_i) = \bot$. We claim that C' is a solution. Firstly, note that $|C'| = 2N = k$. By construction, we know that for every $a \in A'$ we have $u_a(C') = u_a(C \setminus C')$. Finally, for each $i \in \{1,\ldots,M\}$, we have that $u_{d^i}(C') \geq u_{d_i}(C \setminus C')$ if and only if the number of literals in K_i evaluated to $\top$ by f is at least $|K_i|/2$. The latter holds since f is a solution, and hence the claim follows.

($\Leftarrow$) Let C' be a solution. Let $f \colon X \to \{\top, \bot\}$ be defined by $f(x_i) = \top$ if $x_i \in C'$ and $f(x_i) = \bot$ if $\overline{x_i} \in C'$. Due to Lemma 6, f is well-defined. Moreover, due to Lemma 8, for each clause K_i, at least half of the literals are evaluated to $\top$. □

Clearly, BORDA-SAS is in FPT when parameterized by m and in XP when parameterized by n (see Proposition 2). We have the following relation of k to m:

Lemma 9 ($\star$). *Let I be an instance of* BORDA-SAS *with m items and solution size k. If*

$$
\begin{aligned}
&k < m - \frac{1}{2}\left(\sqrt{2m^2 - 2m + 1} + 1\right) \text{ or} \\
&k \geq \frac{1}{2}\left(\sqrt{2m^2 - 2m + 1} + 1\right),
\end{aligned}
\tag{9}
$$

then I is a trivial no- *or* yes-*instance, respectively.*

By Lemma 9, we know that $k \in \Theta(m)$. Thus, we have the following.

Corollary 5. BORDA-SAS *is in FPT when parameterized by k and when parameterized by $m - k$, and admits a problem kernel of size polynomial in $n + k$.*

6 Discussion

We have reported on a systematic study of the parameterized complexity of several variants of SAS. Putting our results in a broader context, resource allocation problems are studied in AI mainly due to the importance of allocating resources between agents in multiagent systems. As such, better understanding of SAS improves our understanding of resource allocation algorithms and of the possibilities of agreeably allocating goods in the context of SAS. Our specific results advance the understanding of the computational complexity of finding such agreeable allocations. Moreover, with efficient algorithms in relevant special cases, finding a smallest agreeable subset becomes applicable as subroutine for more complex decision-making processes, when the parameter values are known to be bounded.

Concretely, the main message of our paper (cf. Table 1) reads as follows: SAS with utility-based additive preferences is NP-hard and hard for all natural parameters considered in our work—including combinations—excluding the number m of items, for which a trivial brute-force FPT-algorithm but, presumably, no polynomial problem kernel exists. With approval preferences, SAS stays NP-hard yet becomes significantly easier and a polynomial kernel exists (w. r. t. $n+k$). Restricting further to t-Approval and t-Veto enables even more polynomial kernels. Surprisingly, this effect is stronger for t-Veto (recall the dual parameter $m - k$). With Borda utilities, SAS remains NP-hard. In terms of parameterized complexity, BORDA-SAS seems to be very similar to APPROVAL-SAS, possibly with even more positive results; the latter may unfold with our most intriguing open question: Is BORDA-SAS fixed-parameter tractable when parameterized by the number n of agents? Moreover, do the Borda results translate to other scoring vectors?

It is worth to analyze further structural parameters and to extend our study to other preference models (such as submodular preferences [3]). Perhaps the most natural related model for future work may be to consider the multidimensional optimization goal of minimizing the number of items in the agreeable set while maximizing the number of agents who agree on it.

Acknowledgements

This research project started at the annual research retreat of the Algorithmics and Computational Complexity (AKT) group of TU Berlin, held in Darlingerode, Germany, March 19–23, 2018. We thank the anonymous reviewers from IJCAI'23 and ECAI'23 for their constructive feedbacks. T. Fluschnik acknowledges support by the DFG, project AFFA (BR 5207/1 and NI 369/15).

References

[1] Faisal N. Abu-Khzam, 'A kernelization algorithm for d-hitting set', *Journal of Computer and System Sciences*, **76**(7), 524–531, (2010).

[2] Haris Aziz, Serge Gaspers, Simon Mackenzie, and Toby Walsh, 'Fair assignment of indivisible objects under ordinal preferences', in *Proceedings of the 2014 International conference on Autonomous Agents and Multi-Agent Systems (AAMAS'14)*, pp. 1305–1312. IFAAMAS/ACM, (2014).

[3] Nawal Benabbou, Mithun Chakraborty, Ayumi Igarashi, and Yair Zick, 'Finding fair and efficient allocations when valuations don't add up', in *Proceedings of the 13th International Symposium on Algorithmic Game Theory (SAGT'20)*, volume 12283 of *LNCS*, pp. 32–46. Springer, (2020).

[4] Nadja Betzler, Jiong Guo, Christian Komusiewicz, and Rolf Niedermeier, 'Average parameterization and partial kernelization for computing medians', *Journal of Computer and System Sciences*, **77**(4), 774–789, (2011).

[5] Nadja Betzler, Rolf Niedermeier, and Gerhard J. Woeginger, 'Unweighted coalitional manipulation under the Borda rule is NP-hard', in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, pp. 55–60. IJCAI/AAAI, (2011).

[6] René van Bevern, 'Towards optimal and expressive kernelization for d-hitting set', *Algorithmica*, **70**(1), 129–147, (2014).

[7] Bernhard Bliem, Robert Bredereck, and Rolf Niedermeier, 'Complexity of efficient and envy-free resource allocation: Few agents, resources, or utility levels', in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*, pp. 102–108. IJCAI/AAAI Press, (2016).

[8] Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo, 'Kernel bounds for disjoint cycles and disjoint paths', *Theoretical Computer Science*, **412**(35), 4570–4578, (2011).

[9] Sylvain Bouveret, Ulle Endriss, and Jérôme Lang, 'Fair division under ordinal preferences: Computing envy-free allocations of indivisible goods', in *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI'10)*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pp. 387–392. IOS Press, (2010).

[10] Steven J. Brams, D. Marc Kilgour, and Christian Klamler, 'The undercut procedure: an algorithm for the envy-free division of indivisible items', *Soc. Choice Welf.*, **39**(2-3), 615–631, (2012).

[11] Robert Bredereck, Jiehua Chen, Piotr Faliszewski, Jiong Guo, Rolf Niedermeier, and Gerhard J. Woeginger, 'Parameterized algorithmics for computational social choice: Nine research challenges', *Tsinghua Science and Technology*, **19**(4), 358–373, (2014).

[12] Robert Bredereck, Till Fluschnik, and Andrzej Kaczmarczyk, 'When votes change and committees should (not)', in *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI'22)*, pp. 144–150. ijcai.org, (2022).

[13] Robert Bredereck, Andrzej Kaczmarczyk, Dusan Knop, and Rolf Niedermeier, 'High-multiplicity fair allocation: Lenstra empowered by n-fold integer programming', in *Proceedings of the 2019 ACM Conference on Economics and Computation (EC'19)*, pp. 505–523. ACM, (2019).

[14] Robert Bredereck, Andrzej Kaczmarczyk, and Rolf Niedermeier, 'Envy-free allocations respecting social networks', *Artificial Intelligence*, **305**, 103664, (2022).

[15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh, *Parameterized Algorithms*, Springer, 2015.

[16] Jessica Davies, George Katsirelos, Nina Narodytska, Toby Walsh, and Lirong Xia, 'Complexity of and algorithms for the manipulation of Borda, Nanson's and Baldwin's voting rules', *Artificial Intelligence*, **217**, 20–42, (2014).

[17] Palash Dey, Neeldhara Misra, and Y. Narahari, 'Kernelization complexity of possible winner and coalitional manipulation problems in voting', *Theoretical Computer Science*, **616**, 111–125, (2016).

[18] Michael Dom, Daniel Lokshtanov, and Saket Saurabh, 'Kernelization lower bounds through colors and ids', *ACM Transactions on Algorithms*, **11**(2), 13:1–13:20, (2014).

[19] Britta Dorn and Ildikó Schlotter, 'Having a hard time? Explore parameterized complexity!', in *Trends in Computational Social Choice*, ed., U. Endriss, chapter 11, 209–230, AI Access, (2017).

[20] Rodney G. Downey and Michael R. Fellows, *Fundamentals of Parameterized Complexity*, Texts in Computer Science, Springer, 2013.

[21] Eduard Eiben, Robert Ganian, Thekla Hamm, and Sebastian Ordyniak, 'Parameterized complexity of envy-free resource allocation in social networks', *Artificial Intelligence*, **315**, 103826, (2023).

[22] Friedrich Eisenbrand and Robert Weismantel, 'Proximity results and faster algorithms for integer programming using the Steinitz lemma', *ACM Transactions on Algorithms*, **16**(1), 5:1–5:14, (2020).

[23] Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette, 'On the parameterized complexity of multiple-interval graph problems', *Theoretical Computer Science*, **410**(1), 53–61, (2009).

[24] Till Fluschnik, Piotr Skowron, Mervin Triphaus, and Kai Wilker, 'Fair knapsack', in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI'19)*, pp. 1941–1948. AAAI Press, (2019).

[25] Lance Fortnow and Rahul Santhanam, 'Infeasibility of instance compression and succinct PCPs for NP', *J. Comput. Syst. Sci.*, **77**(1), 91–106, (2011).

[26] Laurent Gourvès, 'Agreeable sets with matroidal constraints', *Journal of Combinatorial Optimization*, **37**(3), 866–888, (2019).

[27] Danny Harnik and Moni Naor, 'On the compressibility of NP instances and cryptographic applications', *SIAM Journal on Computing*, **39**(5), 1667–1713, (2010).

[28] Pasin Manurangsi and Warut Suksompong, 'Computing an approximately optimal agreeable set of items', in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, pp. 338–344. ijcai.org, (2017).

[29] Pasin Manurangsi and Warut Suksompong, 'Computing a small agreeable set of indivisible items', *Artificial Intelligence*, **268**, 96–114, (2019).

[30] Neeldhara Misra, Arshed Nabeel, and Harman Singh, 'On the parameterized complexity of minimax approval voting', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*, p. 97–105. IFAAMAS, (2015).

[31] Nhan-Tam Nguyen, Dorothea Baumeister, and Jörg Rothe, 'Strategy-proofness of scoring allocation correspondences for indivisible goods', *Soc. Choice Welf.*, **50**(1), 101–122, (2018).

[32] Krzysztof Pietrzak, 'On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems', *J. Comput. Syst. Sci.*, **67**(4), 757–771, (2003).

[33] Warut Suksompong, 'Assigning a small agreeable set of indivisible items to multiple players', in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*, pp. 489–495. IJCAI/AAAI Press, (2016).

[34] Yongjie Yang and Dinko Dimitrov, 'Group control for procedural rules: Parameterized complexity and consecutive domains', *CoRR*, **abs/2203.16872**, (2022).