

Parameterized Algorithms for Finding a Collective Set of Items

Robert Brederick,¹ Piotr Faliszewski,² Andrzej Kaczmarczyk,¹ Dušan Knop,^{1,3} Rolf Niedermeier¹

¹Technische Universität Berlin, Chair of Algorithmics and Computational Complexity

²AGH University, Kraków, Poland

³Department of Theoretical Computer Science, Faculty of Information Technology,
Czech Technical University in Prague, Prague, Czech Republic

{robert.bredereck, a.kaczmarczyk, rolf.niedermeier}@tu-berlin.de, faliszew@agh.edu.pl, dusan.knop@fit.cvut.cz

Abstract

We extend the work of Skowron et al. (AIJ, 2016) by considering the parameterized complexity of the following problem. We are given a set of items and a set of agents, where each agent assigns an integer utility value to each item. The goal is to find a set of k items that these agents would collectively use. For each such collective set of items, each agent provides a score that can be described using an OWA (ordered weighted average) operator and we seek a set with the highest total score. We focus on the parameterization by the number of agents and we find numerous fixed-parameter tractability results (however, we also find some $W[1]$ -hardness results). It turns out that most of our algorithms even apply to the setting where each agent has an integer weight.

1 Introduction

We study the parametrized complexity of the problem of choosing a collective set of items, introduced by Skowron et al. (2016). In this problem we are given a set of agents and a set of items, and the goal is to choose a set of k items that the agents would jointly use. Depending on the setting, each agent may be able to access each of the chosen items, or only one of them (e.g., the one he or she likes most), or some subset of them (e.g., the bottom half of the items, from the perspective of the agent), or even the agents may be uncertain as to which items will eventually be available. Thus, each agent has an intrinsic utility for gaining access to each of the items, but the satisfaction (or, the utility) that the agent derives from a set of items is computed as an *ordered weighted average (OWA)* of these utilities. We seek a set of k items whose sum of thus-computed satisfactions is highest.

An OWA operator $\Lambda = (\lambda_1, \dots, \lambda_k)$ is a vector of rational numbers; to apply Λ to a vector $u = (u_1, \dots, u_k)$ of utilities, we (1) sort u in nonincreasing order—so we obtain a vector $u' = (u'_1, \dots, u'_k)$ —and (2) output the value $\lambda_1 u'_1 + \lambda_2 u'_2 + \dots + \lambda_k u'_k$.

Due to the expressive power of OWA operators, the problem of choosing a collective set of items is very general and captures many interesting settings. Below we first discuss a direct application, and then argue that it also extends the framework of multiwinner voting (Faliszewski et al. 2017b).

Student Recruiting Setting. Consider the task of choosing PhD students for a university department. Each of the professors assigns a utility to each of the prospective candidates (e.g., depending on the results of their entry exams, his or her impressions after interviews, or the students' interests). Then the professors evaluate potential batches of k students. If each professor thought that he or she would get the best PhD student from the batch (with respect to his or her preferences), and he or she is only interested in advising one new student, then the professors should use OWA-vectors $(1, 0, \dots, 0)$. Similarly, if each of the professors thought that he or she would get one of the top two students (with respect to his or her preferences), then they would have a choice between several OWAs. If they viewed the student-professor matching process as effectively random (but restricted to the top two students for each professor), then they should use OWA $(1/2, 1/2, 0, \dots, 0)$.¹ Yet, if they were risk-averse and thought they would get the second-best student, they should use OWA $(0, 1, 0, \dots, 0)$. Other OWA-vectors capture numerous other approaches how the professors might view the matching process.

Interestingly, sometimes it even makes sense to use OWA-vectors with negative values. Let us say that each professor assigns utility 1 to each student whose interests are aligned with his or hers, and utility 0 to each other one. If the student's and the professor's interests are aligned, then the student is likely to require some attention from the professor. As the professors are somehow busy, they might want to limit the number of students who share their interests (although they certainly would like *some* such students to be admitted). For example, OWA-vector $(1, 1, 0, -1, -1, -1, \dots)$ indicates that the professor is happy with one student, is even happier with two, and is as happy with three, but beyond that, his or her satisfaction drops (so, for example, with six students he or she is less satisfied than without any students at all).

Multiwinner Voting Setting. In the multiwinner setting we have a set of voters (agents) who express preferences regarding a set of candidates (items), and the goal is to find a committee (a *size- k* set of items) that, in the aggregate, the voters find most appealing. By choosing the right OWA vectors and setting appropriate agent utilities, we

¹Mathematically, this is equivalent to OWA $(1, 1, 0, \dots, 0)$.

can capture a number of multiwinner voting rules, including k -Borda, Chamberlin–Courant (CC), PAV, and many other ones, including all OWA-based committee scoring rules and all Thiele rules; we provide references and definitions for these rules in Section 2.

To capture the connection with the multiwinner setting, we pay special attention to two families of (intrinsic) utility functions, approval utilities and Borda utilities. Under the former, each agent assigns a 0/1 utility to each of the items, whereas under the latter each agent ranks the items from best to worst and assigns them linearly decreasing values.

Our Contributions. We study the parameterized complexity of finding a collective set of items depending on the nature of the utility functions and the OWA vectors used. We focus on the parameterization by the number of agents, sometimes supplied with parameters regarding the structure of the OWA-vector (parameterization by the number of items leads to a simple, brute-force FPT² algorithm for all settings, whereas parameterization by the size of the collective set yields W[2]-hardness even in the simplest settings). Indeed, in our student recruiting example we would expect to have many more students than professors (the agents). Our main contributions are as follows:

1. We show an FPT algorithm for the case of arbitrary utilities and nonincreasing OWA-vectors that are constant except for the first or last π values (for OWA-vectors that are not nonincreasing we show W[1]-hardness). Our algorithm is parametrized by the number of agents and π . So, when π is fixed, which is common for many natural OWA-vectors, the algorithm is actually in FPT with respect to the number of agents.
2. For the case of approval utilities, we give FPT algorithms for a very broad set of OWA-vectors. However, we also find an interesting OWA-vector (with negative values), for which we obtain W[1]-hardness.
3. We show that whenever we have an FPT algorithm for approval utilities, we also have an FPT approximation scheme for Borda utilities (and some other ones).

In the multiwinner interpretation, our results mean that for Thiele rules (i.e., for the case of approval utilities) we typically have FPT algorithms, but for committee scoring rules that are non-Thiele (e.g., that are based on the Borda utilities) such algorithms are mostly missing, unless the underlying OWA-vector is of quite a special form or one accepts approximate solutions. We present a summary of our results in Table 1. Due to restricted space, we defer many proofs details to a full version of this paper.

²An FPT algorithm with parameter k solves a problem instance I in $f(k) \cdot |I|^{O(1)}$ time, where $f(k)$ may be an arbitrary computable function. While an FPT algorithm can solve a problem instance for constant k in polynomial time where the degree of the polynomial is independent of k , this is not the case for an XP algorithm which merely gives polynomial time for constant k . Finally, W[1]-hardness indicates strong complexity-theoretic evidence that there is no FPT algorithm.

OWA-vector	utilities	complexity	ref.
$(1, 0, \dots, 0)$	any	FPT(n)	BSU
$(\underbrace{1, \dots, 1}_\pi, 0, \dots, 0)$	any	FPT(n, π) XP(n)	Prop. 2 Prop. 8
$(\lambda_1, \dots, \lambda_\pi, 0, \dots, 0)$, for $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_\pi$	any	FPT(n, π)	Prop. 2
$(\lambda_1, \dots, \lambda_k)$, for $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k$	any	FPT(n, k)	Thm. 1
$(0, \dots, 0, 1)$	approval	FPT(n)	Prop. 4
$(0, 1, 0, \dots, 0)$	approval	FPT(n)	Thm. 5
OWA-vector has ρ different values	approval	FPT(n, ρ)	Thm. 5
OWA is piecewise non-increasing with ρ pieces	approval	FPT(n, ρ)	Thm. 5
$(1, 2, 3, \dots, k)$	approval	FPT(n)	Thm. 6
$(\lambda_1, \dots, \lambda_k)$ $\lambda_i \in \{-1, 0, 1\}, \forall i \in [k]$	approval	W[1]-h.(n)	Thm. 7
$(0, 1, 0, \dots, 0)$	Borda	W[1]-h.(n) FPT-AS(n)	Thm 3, Thm. 10
$(1, 1/2, 1/3, \dots, 1/k)$ $(\lambda_1, \dots, \lambda_\pi, 0, \dots, 0)$, for $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_\pi$	Borda	FPT-AS	Thm. 10

Table 1: Examples of the parametrized complexity of OWA-WINNER (it is NP-hard for all the listed settings; for vectors $(1, 0, \dots, 0)$ this is due to Procaccia et al. (2008) and Lu and Boutilier (2011), for vectors $(1, 2, 3, \dots)$ this is due to Faliszewski et al. (2018b), and for other vectors this is due to Skowron et al. (2016)). The result marked BSU is due to Betzler et al. (2013).

2 Preliminaries

We first specify our notation, then formalize the problem of choosing a collective set of items, and relate it to voting.

Notation

For a positive integer t , we write $[t]$ to refer to the set $\{1, \dots, t\}$. We use the Iverson bracket notation, that is, for a logical expression F , we write $[F]$ to denote 1 if F is true and to denote 0 otherwise. For a vector $x = (x_1, \dots, x_n)$, we sometimes write $x[i]$ to refer to x_i (this is particularly convenient when we discuss a vector x without listing its components explicitly).

Choosing a Collective Set of Items

Let $N = [n]$ be a set of agents and let $R = \{r_1, r_2, \dots, r_m\}$ be a set of items (resources). A preference function (or, a utility function) is a function $u: R \rightarrow \mathbb{N}$ that assigns numerical preferences (utilities) to the items. A preference profile is a collection $U = \{u_1, u_2, \dots, u_n\}$ of preference functions over the same set of items, with one function for each agent. For a given item r_j and a given agent i , we say that $u_i(r_j)$ is the intrinsic utility of r_j for agent i .

Special Preference Functions. Certain families of preference functions are particularly interesting because they allow us to connect our results to the literature on multiwinner voting (Faliszewski et al. 2017b) and budgeted social choice (Lu and Boutilier 2011). Specifically, we consider Borda preferences and approval preferences:

1. A *Borda preference function* for a set R of m items is a bijection $u: R \rightarrow [m-1] \cup \{0\}$. Intuitively, an agent has a ranking of the items and he or she assigns to each item the utility equal to the number of items ranked below. We often specify Borda preferences through *preference orders*. For example, by writing $i: r_1 \succ r_2 \succ \dots \succ r_m$, we mean that agent i assigns utility $m-1$ to item r_1 , utility $m-2$ to item r_2 , and so on.
2. An *approval preference function* is a function that for each item assigns either utility 1 or 0 (the former means that the item is *approved* and the latter that it is *disapproved*). Sometimes we consider *weighted approval preference functions*, which for each item assign either some positive constant q or 0 (when we speak of profiles of weighted approval functions, then each agent a may have his or her individual value $q(a)$).

Ordered Weighted Average Operators (OWA). We refer to sets of items as *item sets*, and to size- k item sets as *k -item sets*. For a k -item set $S = \{s_1, s_2, \dots, s_k\}$ and preference function u , we write $\tilde{u}(S)$ to denote the vector of utilities that u assigns to the items from S , sorted in nonincreasing order. Similarly, we write $\hat{u}(S)$ to denote a vector of items from S ordered nonincreasingly with respect to their utilities (ties are broken lexicographically).

Example 1. Let $R = \{r_1, r_2, \dots, r_{10}\}$, $S = \{r_1, r_2, r_4\}$ and u be such that $u(r_1) = 10$, $u(r_2) = 15$, $u(r_3) = 9$, $u(r_4) = 10$, etc. We have $\tilde{u}(S) = (15, 10, 10)$ and $\hat{u}(S) = (r_2, r_1, r_4)$.

For a given positive integer k , let $\Lambda_k = (\lambda_1, \lambda_2, \dots, \lambda_k)$ be a vector of rational numbers. We define the Λ_k -score of a k -item set $S = \{s_1, s_2, \dots, s_k\}$ according to preference function u as $\Lambda_k\text{-score}_u(S) := \sum_{j=1}^k \lambda_j (\tilde{u}(S)[j])$. We extend this notion to preference profiles as follows. For a preference profile $U = \{u_1, u_2, \dots, u_n\}$, we have:

$$\Lambda_k\text{-score}_U(S) := \sum_{i=1}^n \Lambda_k\text{-score}_{u_i}(S).$$

Sometimes, when the context is clear, instead of writing $\Lambda_k\text{-score}_{u_i}(\cdot)$, we write $\Lambda_k\text{-score}_i(\cdot)$. We refer to Λ_k as the *OWA-vector* and to $\Lambda_k\text{-score}$ as the *OWA-score* (OWA stands for *Ordered Weighted Average*; operators of this form were introduced by Yager (1988)).

Remark 1. One may wonder why we speak of an *OWA-score* and not *OWA-utility*. We do so because we want our notation to be compatible with that from the multiwinner voting literature, where speaking of the scores of item sets (committees) is typical.

Main Problem. We study the problem of finding a k -item set with the highest possible OWA-score, originally introduced by Skowron et al. (2016).

Definition 1. In the OWA-WINNER problem, we are given a set $N = [n]$ of agents, a set R of m items, a preference profile U (consisting of n preference profiles), a positive integer $k < m$, an OWA-vector Λ_k , and an integer T . We ask whether there exists a k -item set $S \subseteq R$ such that $\Lambda_k\text{-score}_U(S) \geq T$.

Relation to Multiwinner Voting

Our setting is very close to that of multiwinner voting. It is beyond the scope of this paper to describe the multiwinner framework in detail, but below we express some well-known voting rules in the language of the OWA-WINNER problem:

1. For Borda utilities and OWA-vectors of the form $(1, \dots, 1)$, the OWA-WINNER problem is equivalent to winner determination under the k -Borda rule. For the same OWA-vector and approval utilities, we obtain the multiwinner approval rule.
2. For Borda utilities and OWA-vectors of the form $(1, 0, \dots, 0)$, we obtain the Chamberlin–Courant rule, β -CC (Chamberlin and Courant 1983). For approval utilities, we get approval-based Chamberlin–Courant rule, α -CC.
3. For approval utilities and OWA-vectors of the form $(1, 1/2, \dots, 1/k)$ we get the proportional approval voting rule, PAV (Kilgour 2010; Thiele 1895).

The literature regarding the computational complexity of these rules (and, thus, these variants of our problem) is quite rich. While for k -Borda and multiwinner approval voting there are simple polynomial-time algorithms, almost all other settings lead to NP-hardness. Indeed, this is the case for each of α -CC (Procaccia, Rosenschein, and Zohar 2008), β -CC (Lu and Boutilier 2011), and PAV (Skowron, Faliszewski, and Slinko 2015; Aziz et al. 2015). Yet, there are various workarounds for these hardness results, including approximation algorithms (Lu and Boutilier 2011; Skowron, Faliszewski, and Lang 2016; Skowron, Faliszewski, and Slinko 2015), heuristics (Faliszewski et al. 2018c; 2018a), and algorithms for restricted preference domains (Betzler, Slinko, and Uhlmann 2013; Skowron et al. 2015; Peters 2018).

For the case of $\{\alpha, \beta\}$ -CC and PAV there are also FPT algorithms (Betzler, Slinko, and Uhlmann 2013; Faliszewski et al. 2018b) parameterized by the number of voters (agents). We extend these results as our framework incorporates the class of OWA-based committee scoring rules (Elkind et al. 2017; Faliszewski et al. 2019), and the class of Thiele rules (Lackner and Skowron 2018; Aziz et al. 2015); e.g., β -CC is an OWA-based committee scoring rule and PAV is a Thiele rule. In our language, winner determination problems for OWA-based committee scoring rules correspond to OWA-WINNER where all agents’ utility functions are identical up to permuting items. Thiele rules are modeled by assuming that all agents have approval utilities.

3 Arbitrary Utilities

We start our discussion by considering the setting where the utilities are unrestricted, but the OWA-vectors are required

to be nonincreasing. In this case we get an FPT algorithm for a joint parameterization by the number of agents and the size of the item set. Then we use this algorithm to get FPT algorithms for more restricted OWAs, but with more useful parameterizations.

Theorem 1. *There is an FPT algorithm for OWA-WINNER with nonincreasing OWA vectors, parameterized by the number of agents and the size of the item set.*

Proof. The input for our algorithm consists of the set of items $R = \{r_1, r_2, \dots, r_m\}$, preference profile $U = \{u_1, u_2, \dots, u_n\}$, OWA vector $\Lambda_k = (\lambda_1, \dots, \lambda_k)$ such that $\lambda_1 \geq \dots \geq \lambda_k$, and the score threshold T .

We first describe our ideas intuitively. Let $S = \{s_1, \dots, s_k\}$ be a target k -item set. For each agent i and each integer $j \in [k]$, we define $\psi_i(j)$ to be ℓ if for agent i , item s_j is the ℓ -th most-valued one within those in S (breaking ties lexicographically). More formally, $\psi_i(j) = \ell$ if $\hat{u}_i(S)[\ell] = s_j$. Note that each ψ_i is a permutation of $[k]$. Further, let $h: [k] \rightarrow R$ be a function assigning each $j \in [k]$ a different item from R . (We refer to h as the matching between S and items from R .) Using this notation, we express the OWA-score of S as:

$$\Lambda_k\text{-score}_U(S) = \sum_{i=1}^n \sum_{j=1}^k \lambda_{\psi_i(j)} u_i(h(j)). \quad (1)$$

In other words, to compute the score of S we use the following components:

1. a matching h of the elements from S to items in R (this moves us from the domain of items to the domain of their indices within S), and
2. permutations ψ_1, \dots, ψ_n , which describe orders $\hat{u}_i(S)$, but using the indices within S instead of item names.

Note that h implicitly defines S ; the matched items are in S , the other ones are not. The main idea of our algorithm is to guess the permutations ψ_1, \dots, ψ_n and then find a matching h (of the numbers from $[k]$ to items) that maximizes the score computed with Eq. (1). Then we reconstruct S using h .

We now describe our algorithm formally. First, for each agent i , we guess a permutation $\psi_i: [k] \rightarrow [k]$. Second, for each item $r \in R$ and each number $j \in [k]$ we let:

$$w(r, j) = \sum_{i=1}^n \lambda_{\psi_i(j)} u_i(r).$$

Then we form a complete bipartite graph, which has the integers from $[k]$ on the left and the items from R on the right. For each item $r \in R$ and each number $j \in [k]$, we set the weight of the edge between them to be $w(r, j)$. Finally, we find a maximum-weight matching in this graph.

Let $h: [k] \rightarrow R$ be such a matching with the highest weight over all possible guesses of ψ_1, \dots, ψ_n . We return “yes” if the weight of h is at least T .

Next, we show the algorithm’s correctness. Let $W = \{h(1), \dots, h(k)\}$ be the set corresponding to h and let ψ_1, \dots, ψ_n be the permutations that lead to it. If for each agent i and each two distinct integers $j, j' \in [k]$ it holds that $\psi_i(j) < \psi_i(j')$ implies that $h(j)$ is ahead of $h(j')$ in $\hat{u}_i(W)$, then the weight of matching h in our bipartite graph equals the score of W . Indeed, in this case for each agent i we have:

$$\hat{u}_i(W) = (h(\psi_i^{-1}(1)), h(\psi_i^{-1}(2)), \dots, h(\psi_i^{-1}(k))).$$

By the choice of h as the maximum-weight matching over the choice of ψ_1, \dots, ψ_n , W is a highest-scoring item set. Indeed, the item set S analyzed in the intuitive explanation leads to a matching with the same weight.

On the contrary, if there is an agent i and two numbers $j, j' \in [k]$ such that $\psi_i(j) < \psi_i(j')$ but in $\hat{u}_i(W)$ item $h(j')$ is ahead of $h(j)$, then we can transform ψ_i , so this inversion is removed, without decreasing the weight of h . Indeed, if there is such an inversion, then we simply swap the values of $\psi_i(j)$ and $\psi_i(j')$ (and update the weights of the edges in our bipartite graph). Since $\lambda_1 \geq \dots \geq \lambda_k$ (and because $\hat{u}_i(\cdot)$ sorts items nonincreasingly with respect to their values for agent i), it must be the case that the new weight of h does not decrease. Further, by the choice of h and permutations ψ_1, \dots, ψ_n so that the original weight of h was maximum, it also does not increase. We repeat this process until we remove all inversions. \square

We prove the next result using the same approach, but with different weights in the constructed bipartite graph.

Proposition 2. *There is an FPT algorithm, parameterized by the number of agents and π , for OWA-WINNER with nonincreasing OWAs that are constant except for either the top π or the bottom π values.*

Theorem 1 and Proposition 2 provide a broad generalization of the FPT algorithms for α -CC and β -CC, parameterized by the number of agents (Betzler, Slinko, and Uhlmann 2013). Indeed, for constant values of π our algorithms run in FPT time with respect to the number of voters. Since CC rules use OWAs $(1, 0, \dots, 0)$, Proposition 2 applies. Yet, if we were interested in vectors of the form $(0, 1, 0, \dots, 0)$, then the situation would be quite different. Let us consider the following variant of the OWA-WINNER problem.

Definition 2. *Let i be an integer. By $\beta(i)$ -WINNER, we mean the OWA-WINNER problem with Borda preferences and OWA-vector of the form $(\underbrace{0, \dots, 0}_{i-1}, 1, 0, \dots, 0)$.*

$\beta(1)$ -WINNER models winner determination under the β -CC multiwinner voting rule. There is also a number of settings captured by $\beta(i)$ -WINNER with $i > 1$ (see Section 1 and the work of Skowron et al. (2016)). Yet, for $i > 1$ Proposition 2 does not apply and, in fact, we show $W[1]$ -hardness.

Theorem 3. *$\beta(2)$ -WINNER is $W[1]$ -hard when parameterized by the number of agents.*

Proof Sketch. We reduce from the $W[1]$ -complete problem MULTICOLORED CLIQUE (parameterized by the solution size). An instance of this problem consists of a graph $G = (V(G), E(G))$, where each vertex has one of h colors, and we ask if there are h vertices such that each of them has a different color and is connected to every other one.

Let G be our input graph and let h be the number of colors ($h \geq 7$). Without loss of generality, there are exactly q vertices of each color. Hence, for each $i \in [h]$, let $V^{(i)} = \{v_1^{(i)}, \dots, v_q^{(i)}\}$ be the set of vertices of color i . For each two distinct $i, j \in [h]$ let $E^{(i,j)}$ be the set of edges that connect vertices of color i with those of color j ; note that

$E^{(i,j)} = E^{(j,i)}$. Further, we assume that there are no edges between vertices of the same color. For each color $j \in [h]$ and each vertex $v \in V(G)$, by $E_j(v)$ we mean the set of edges that connect v with vertices of color j . We assume that for each $j \in [h]$ and each $v \in V(G)$ we have $|E_j(v)| \leq q-1$ (if not, then we add one new vertex of each color). We set $H = \binom{h}{2}$.

We form an instance of $\beta(2)$ -WINNER as follows. Let the set of items be $R = V(G) \cup E(G) \cup D$, where $D = \{d_1, d_2, d_3, \dots\}$ is a set of dummy items (later it will become clear we have polynomially many of them). Let:

$$L = 4hq + 4Hq^2 + 2(h-1)^2q^2$$

be a certain large number (its choice will become clear later).

As the agents have Borda utilities, we describe them using preference orders. Whenever in a preference order we write $D(\ell)$, where ℓ is some positive integer, we mean listing ℓ new dummy items that do not appear among the top L items in the other preference orders (this implicitly extends the set D). Whenever we write *others* in a preference order, we mean listing the remaining items in some arbitrary order. Each time we write a set Y in a preference order, we mean an arbitrary but fixed order of elements in Y ; by \overrightarrow{Y} we denote the inverse order of that for Y .

Let $k = h + H$ be the item set size. Our intention is that if there is a multicolored clique, then each highest-scoring item set consists of h vertex items and H edge items that form a clique. We form the following groups of agents:

1. For each pair of colors $1 \leq i < j \leq h$, *vertex selection group* contains agents $v_{i,j}$ and $v'_{i,j}$ with preference orders:

$$\begin{aligned} v_{i,j}: V^{(i)} \succ V^{(j)} \succ D(L-2q) \succ \text{others}, \\ v'_{i,j}: \overrightarrow{V^{(i)}} \succ \overrightarrow{V^{(j)}} \succ D(L-2q) \succ \text{others}. \end{aligned}$$

2. The *edge selection group* is defined similarly, but to specify it formally we need additional notation. We let \mathcal{E} be a sequence of all sets of the form $E^{(i,j)}$, where $1 \leq i < j \leq h$, such that $\mathcal{E} = E^{(1,2)}, E^{(1,3)}, \dots, E^{(1,h)}, E^{(2,3)}, \dots$. We write $\mathcal{E}(\ell)$ to refer to the ℓ -th set in this sequence. Note that \mathcal{E} has exactly $H = \binom{h}{2}$ elements. By $\mathcal{E}'(\ell)$ we mean the partial preference order of the form $\mathcal{E}(\ell) \succ D(q^2 - |\mathcal{E}(\ell)|)$; each occurrence of $\mathcal{E}'(\ell)$ in a preference order uses a fresh set of dummy items. For each $1 \leq \ell < \ell' \leq H$, the edge selection group contains two agents $e_{\ell,\ell'}$ and $e'_{\ell,\ell'}$ with the following preference orders:

$$\begin{aligned} e_{\ell,\ell'}: \mathcal{E}'(\ell) \succ \mathcal{E}'(\ell') \succ D(L-2q^2) \succ \text{others}, \\ e'_{\ell,\ell'}: \overrightarrow{\mathcal{E}'(\ell)} \succ \overrightarrow{\mathcal{E}'(\ell')} \succ D(L-2q^2) \succ \text{others}. \end{aligned}$$

3. To form the *consistency group*, we use the following notation. Let $v \in V(G)$ be a vertex. For a color $j \in [h]$, by $S_j(v)$ we mean the partial preference order $E_j(v) \succ D(q-1 - |E_j(v)|) \succ v$. For each pair of colors $i, j \in [h]$, $i \neq j$, we have agents $c_{i,j}$ and $c'_{i,j}$ with preference orders:

$$\begin{aligned} c_{i,j}: S_j(v_1^{(i)}) \succ \dots \succ S_j(v_q^{(i)}) \succ D(L-q^2) \succ \text{others}, \\ c'_{i,j}: S_j(v_q^{(i)}) \succ \dots \succ S_j(v_1^{(i)}) \succ D(L-q^2) \succ \text{others}. \end{aligned}$$

(Recall the intended form of a highest-scoring item set for the case where our clique exists. For such an item set X , the agents $c_{i,j}$ and $c'_{i,j}$ would assign to X score equal to the (Borda) utility of the selected item from $V^{(i)}$.)

Let m denote the total number of items in our instance. We define the desired score value to be:

$$\begin{aligned} T = 2H(m - 3/2q - 1/2) + 2\binom{H}{2}(m - 3/2q^2 - 1/2) \\ + 2(h-1)^2(m - q/2 - q^2/2). \end{aligned}$$

We claim that there is an item set with score at least T if and only if G contains a multicolored clique of size h .

We say that an item set is *proper* if for each color $i \in [h]$ it contains a single vertex from $V^{(i)}$, and for each pair of colors $1 \leq i < j \leq h$, it contains a single edge from $E^{(i,j)}$. Let X be a proper item set. We will now calculate the score of X . We will see that if X corresponds to a multicolored clique, then it has score T ; otherwise its score is lower. Non-proper item sets have even lower scores.

Observe that each pair of vertex selection agents assigns exactly $2m - 3q - 1$ points to X . Indeed, for each two agents $v_{i,j}$ and $v'_{i,j}$, the second highest-ranked member of X is some vertex item $v \in V^{(j)}$; one of the agents assigns to v utility of the form $m - q - t$ and the other one assigns $m - q - (q - t + 1)$, where t is an integer in $[q]$. Altogether, this gives score $2m - 3q - 1$. Similarly, each pair of edge selection agents assigns utility $2m - 3q^2 - 1$ to X . In total, the vertex and edge selection agents assign to X score:

$$\begin{aligned} H(2m - 3q - 1) + \binom{H}{2}(2m - 3q^2 - 1) \\ = 2H(m - 3/2q - 1/2) + 2\binom{H}{2}(m - 3/2q^2 - 1/2). \end{aligned}$$

The agents from the consistency group assign scores as follows. Let us consider two arbitrary colors $i, j \in [h]$, $i \neq j$. Let u and e be the respective unique members of $V^{(i)}$ and $E^{(i,j)}$ included in X . If u is incident to e , then both agents $c_{i,j}$ and $c'_{i,j}$ rank e ahead of u . Thus u is the second highest-ranked member of X according to these agents and they assign score $m - q + m - q^2 = 2m - q - q^2$ to X . Yet, if u is not incident to e , then u is the second highest-ranked member of X according to exactly one of $c_{i,j}$ and $c'_{i,j}$, and e is the second highest-ranked member of X according to the other one. Then these agents assign score strictly lower than $2m - q - q^2$.

In summary, if X contains vertices and edges that form a multicolored clique, then its score is T . Otherwise, if X is proper but does not correspond to a multicolored clique, then its score is below T . In the full proof we show that if X is not proper, then its score is even lower. \square

Similar proofs show that $\beta(i)$ -WINNER is $W[1]$ -hard for each $i \geq 2$. Indeed, even $\beta(k-1)$ -WINNER is $W[1]$ hard. Yet the complexity of $\beta(k)$ -WINNER remains elusive.

Corollary 1. *For each integer i , $2 \leq i \leq k-2$, it holds that the $\beta(i)$ -WINNER and $\beta(k-i)$ -WINNER problems are $W[1]$ -hard for the parameterization by the number of agents.*

4 Approval Utilities

In this section we assume that the agents have (weighted) approval utilities; i.e., effectively we consider Thiele multi-winner rules (Aziz et al. 2015; Lackner and Skowron 2018). We find a broad spectrum of OWA-vectors for which our problem has FPT algorithms (for the parameterization by the number of agents). Indeed, Faliszewski et al. (2018b) have shown such an algorithm for nonincreasing OWAs (in the unweighted case). We extend their result to piecewise nonincreasing OWAs; an OWA-vector consists of ρ nonincreasing pieces if it is of the form:

$$\left(\lambda_1^1, \lambda_1^2, \dots, \lambda_1^{\ell(1)}, \lambda_2^1, \lambda_2^2, \dots, \lambda_2^{\ell(2)}, \dots, \lambda_\rho^1, \lambda_\rho^2, \dots, \lambda_\rho^{\ell(\rho)} \right)$$

where $\ell(1), \dots, \ell(\rho)$ are the sizes of the pieces, and for each $i \in [\rho]$ and each $j \in [\ell(i)-1]$ it holds that $\lambda_i^{j+1} \leq \lambda_i^j$. While piecewise nonincreasing OWA-vectors may seem quite artificial at first, they include a number of interesting cases. For example, an OWA with ρ different values can be seen as piecewise nonincreasing.

We write α -WINNER to refer to OWA-WINNER for the case of approval preferences, and α_{weighted} -WINNER for the case of weighted approval preferences. As a warm-up, let us consider the following proposition.

Proposition 4. *There is an FPT algorithm, parameterized by the number of agents, for α_{weighted} -WINNER with OWAs of the form $(0, \dots, 0, 1)$.*

Proof. For a given item set S , if an agent approves all members of S , then he or she gives it score 1, and otherwise he or she gives it score 0. Thus it suffices to guess a subset of agents of size at least T that would give score 1 to our item set (where T is the given score threshold) and check if the intersection of the sets of items that these agents approve contains at least k items (where k is the size of the item set that we seek). \square

The next theorem is proved using a very broad generalization of the idea from the above result, where guessing the agents is replaced with an integer linear program (solved using Lenstra’s algorithm (Lenstra, Jr. 1983) on top of the trick of Brederick et al. (2017)).

Theorem 5. *There is an FPT algorithm, parameterized by the number of agents and the number ρ , for α_{weighted} -WINNER in case of OWAs that consist of ρ nonincreasing pieces.*

We also show an FPT algorithm for α_{weighted} -WINNER with linearly increasing OWAs; this result is possible due to a recent progress in integer programming.

Theorem 6. *There is an FPT algorithm, parameterized by the number of agents and the number z , for α_{weighted} -WINNER in case of linearly increasing OWAs $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$ with $\lambda_i = x + (i-1)d$ when x, d and the weight $q(a)$ of each agent a are upper-bounded by z .*

Proof. Our input consists of a set R of items, preference profile U , OWA-vector $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$ such that for each $i \in [k]$ we have $\lambda_i = x + (i-1)d$, and score threshold T . Each agent a assigns either utility 0 or $q(a)$ to each

item. We solve this instance by a reducing it to an instance of quadratic integer programming (QIP) with the number of integer variables bounded by a function of the parameter (i.e., the number of agents). We need a quadratic goal function in order to express, for some given number z , the sum of the first z entries of our OWA-vector Λ . It holds that $\sum_{1 \leq i \leq z} \lambda_i = z(x - d/2) + dz^2/2$.

Let N be our set of agents and let $n = |N|$ be the number of agents. We use the following variables:

1. For each subset $N' \subseteq N$, we use variable $x_{N'}$ indicating the number of items in the solution that receive non-zero utility from all agents in N' but zero utility from all other agents. Note that each item is associated with exactly one of the sets N' .
2. For each agent $a \in N$ we use an integer variable y_a indicating the number of items in the solution that receive non-zero utility from a .

The main insight is that every solution can be fully characterized by the $x_{N'}$ variables. We denote by $\#(N')$ the number of items that receive non-zero utility from N' and zero utility from $N \setminus N'$. Our QIP is as follows:

$$\text{maximize } \sum_{1 \leq a \leq N} (y_a(x - d/2) + dy_a^2/2)q(a) \quad (2)$$

$$\text{subject to: } \sum_{N' \subseteq N} x_{N'} = k \quad (3)$$

$$\forall N' \subseteq N: 0 \leq x_{N'} \leq \#(N') \quad (4)$$

$$\forall a \in N: y_a = \sum_{N' \subseteq N, a \in N'} x_{N'} \quad (5)$$

Goal function 2 gives the score of the selected k -item set. Constraint 3 ensures that exactly k items are chosen. Constraints 4 ensure that the solution only consists of items that are indeed available. Constraints 5 ensure that the y_a variables are correctly computed from the $x_{N'}$ variables.

The number of integer variables is upper-bounded by $2^N + N$. All coefficient are upper-bounded by $\max(\{q(a) \mid a \in N\} \cup \{d, x\})$. Hence, fixed-parameter tractability follows from a result of Lokshantov (2015). \square

The two above theorems notwithstanding, there are variants of α -WINNER that are $W[1]$ -hard.

Theorem 7. *α -WINNER is $W[1]$ -hard for the parameterization by the number of agents, even if the OWAs are restricted to contain only -1 s, 0 s and 1 s.*

The general idea is to reduce from the MULTICOLORED CLIQUE problem, as in the proof of Theorem 3, but instead of having a candidate for each vertex in the graph, we encode chosen vertices of a given color with including sufficiently many candidates of a given type in the item set. To make this idea work, it is necessary to form a set of polynomially bounded numbers, so that the sum of each pair is different (Erdős and Turán 1941); this way (and also using consecutive -1 and 1 entries in the OWA) items selected to encode a pair of vertices also encode an edge connecting them.

It is interesting if, parameterized by the number of agents, α -WINNER restricted to OWAs with nonnegative entries is, in general, W[1]-hard or has an FPT algorithm. To answer this open question, it seems that either we need to make progress in solving integer programs, or we need to find a particularly clever trick for building a reduction.

5 Borda Utilities and Approximation

While for (weighted) approval utilities we provided a very broad set of FPT algorithms, for many practical OWA-vectors, Borda utilities seem to be much harder to work with. Indeed, aside from Theorem 1 and Proposition 2, the only exact algorithm for this setting regards OWAs that consist of ones followed by zeros. And even in this case we only get an XP algorithm (but for arbitrary utilities).

Proposition 8. *There is an XP algorithm, parameterized by the number of agents, for OWA-WINNER with OWAs of the form $(1, \dots, 1, 0, \dots, 0)$.*

Indeed, for the case of Borda utilities there are many natural families of OWA-vectors for which we neither have FPT algorithms nor W[1]-hardness proofs, nor even XP algorithms, even though we have FPT algorithms for them for the case of (weighted) approval utilities. For example, this is the case for harmonic OWAs, i.e., OWAs of the form $(1, 1/2, \dots, 1/k)$, used, e.g., in the Harmonic Borda multi-winner rule (Faliszewski et al. 2017a). However, as a consolation prize, typically for settings where we have FPT algorithms for the case of (weighted) approval utilities, we automatically get FPT approximation schemes for Borda utilities (and for many other ones as well). To this end, we need the ability to translate arbitrary utilities to approval ones (we achieve this by replacing each agent with multiple ones, approving more and more items, depending on the utilities that the original agent assigns to them).

Lemma 9. *There is a polynomial-time algorithm that given an instance of OWA-WINNER with n agents, where the highest utility value is u_{\max} , translates it to an equivalent instance with at most $n \cdot u_{\max}$ agents, the same items, and the same OWA (by equivalence, we mean that each item set has the same score in both instances).*

Proof. Consider an OWA-WINNER instance with items $R = \{r_1, \dots, r_m\}$, preference profile U for agents in the set N , and OWA $\Lambda = (\lambda_1, \dots, \lambda_k)$. We form a new instance which is identical, except that we replace each agent $a \in N$, with $\max_{r \in R} u_a(r)$ “approval agents” as follows: We rename the items so that $R = \{r'_1, \dots, r'_m\}$ and $u_a(r'_1) \geq u_a(r'_2) \geq \dots \geq u_a(r'_m)$, and introduce:

$$\begin{aligned} & u_a(r'_1) - u_a(r'_2) \text{ agents who approve } \{r'_1\}, \\ & u_a(r'_2) - u_a(r'_3) \text{ agents who approve } \{r'_1, r'_2\}, \\ & \vdots \\ & u_a(r'_{m-1}) - u_a(r'_m) \text{ agents who approve } \{r'_1, \dots, r'_{m-1}\}, \\ & u_a(r'_m) \text{ agents who approve } \{r'_1, \dots, r'_m\}. \end{aligned}$$

Simple calculations show that for each k -item set S , its scores in both instances are the same. \square

We also need the following definition, which requires that each agent values sufficiently many items sufficiently highly.

Definition 3 (Skowron et al. (2016)). *Consider a setting with m items and let u_{\max} denote the highest utility that some agent gives to an item. Let β and γ be two numbers in $[0, 1]$. The agents have (β, γ) -non-finicky utilities if every agent has utility at least βu_{\max} for at least γm items.*

In particular, Borda utilities are $(0.5, 0.5)$ -non-finicky. We express the next theorem for this class of utilities, but it is possible to extend it to less restricted ones.

Theorem 10. *If there is an FPT algorithm (parameterized by the number of agents) for OWA-WINNER with approval preferences and a given family of non-negative OWAs, then there is an FPT-approximation scheme (parameterized by the number of agents) for OWA-WINNER with $(0.5, 0.5)$ -non-finicky utilities and the same family of OWAs.*

Proof. Let $U = \{u_1, u_2, \dots, u_n\}$ be our preference profile of $(0.5, 0.5)$ -non-finicky utilities of agents in $N = [n]$ over a set R of items, and let k be the size of the desired item set. We use OWA $\Lambda = (\lambda_1, \dots, \lambda_k)$. Let m be the number of items and n be the number of agents. We require that $k \leq m/2$ (but see comments below the proof). We are also given a real number $\varepsilon > 0$ and our goal is to find a size- k item set S with $\Lambda\text{-score}_U(S) \geq (1 - \varepsilon)\text{OPT}$, where $\text{OPT} := \max_{S^* \subseteq R, |S^*|=k} \Lambda\text{-score}_U(S^*)$.

Let t be the smallest positive integer such that $\frac{1}{n^t} \leq \varepsilon$ and let u_{\max} be the highest utility value that an agent assigns to an item. We form a preference profile $U' = \{u'_1, u'_2, \dots, u'_n\}$ so that for each $a \in N$ and $r \in R$:

$$u'_a(r) = \lfloor \frac{u_a(r)}{u_{\max}/n^{t+2}} \rfloor = \lfloor (n^{t+2}) \frac{u_a(r)}{u_{\max}} \rfloor.$$

Let u'_{\max} be the highest utility that an agent assigns to an item in U' . We note that $u'_{\max} \leq n^{t+2}$ and, thus, by Lemma 9 and our assumptions, there is an FPT algorithm for OWA-WINNER with preference profile U' and OWA Λ . Our algorithm proceeds as follows. If $n = 1$, then we simply return the top k items according to the only agent. Otherwise we output a maximum score size- k item set with respect to U' . Below we argue that this item set indeed achieves at least $(1 - \varepsilon)\text{OPT}$ score with respect to U .

We assume that $n \geq 2$. Let S' be a size- k item set that maximizes $\Lambda\text{-score}_{U'}$ and let $S = \{s_1, \dots, s_k\}$ be a size- k item set that maximizes $\Lambda\text{-score}_U$. Naturally, we have that $\Lambda\text{-score}_U(S) = \text{OPT}$ and $\Lambda\text{-score}_{U'}(S') \geq \Lambda\text{-score}_{U'}(S)$. (Note that $\Lambda\text{-score}_{U'}(S) = \sum_{a \in N} \sum_{i=1}^k \lambda_i \check{u}'_a(S)[i]$; we use this and similar equalities below.) We have the following inequality (using that for positive integers x and y : $x \lfloor \frac{y}{x} \rfloor \geq y - x$):

$$\begin{aligned} \Lambda\text{-score}_U(S') &\geq \frac{u_{\max}}{n^{t+2}} \Lambda\text{-score}_{U'}(S') \\ &\geq \frac{u_{\max}}{n^{t+2}} \Lambda\text{-score}_{U'}(S) = \frac{u_{\max}}{n^{t+2}} \sum_{a \in N} \sum_{i=1}^k \lambda_i \check{u}'_a(S)[i] \\ &= \sum_{a \in N} \sum_{i=1}^k \lambda_i \frac{u_{\max}}{n^{t+2}} \lfloor \frac{\check{u}_a(S)[i]}{u_{\max}/n^{t+2}} \rfloor \\ &\geq \sum_{a \in N} \sum_{i=1}^k \lambda_i (\check{u}_a(S)[i] - \frac{u_{\max}}{n^{t+2}}) \\ &= \Lambda\text{-score}_U(S) - \sum_{a \in N} \sum_{i=1}^k \lambda_i \frac{u_{\max}}{n^{t+2}} \end{aligned}$$

$$\begin{aligned}
&= \Lambda\text{-score}_U(S) - \sum_{i=1}^k \lambda_i \frac{u_{\max}}{n^{t+1}} \\
&\geq \Lambda\text{-score}_U(S) - \varepsilon \left(\sum_{i=1}^k \lambda_i \frac{u_{\max}}{n} \right).
\end{aligned}$$

As $n \geq 2$, the utilities are $(0.5, 0.5)$ -non-finicky, $k \leq m/2$, and OPT is at least as high as the highest score that an agent assigns to his or her top k items, we have that $\sum_{i=1}^k \lambda_i \frac{u_{\max}}{n} \leq \sum_{i=1}^k \lambda_i \frac{u_{\max}}{2} \leq \text{OPT}$. Thus we have $\Lambda\text{-score}_U(S') \geq \Lambda\text{-score}_U(S) - \varepsilon \left(\sum_{i=1}^k \lambda_i \frac{u_{\max}}{n} \right) \geq (1 - \varepsilon)\text{OPT}$. This completes the proof. \square

By choosing larger t , we can make the above proof work for (x, x) -non-finicky for any fixed x and for $k \geq m/2$.

6 Conclusions

We have studied the complexity of finding a collective set of items, parameterized by the number of agents. As this framework is very broad, we obtained results that directly apply to multiwinner Thiele rules and to committee scoring rules. For Thiele rules we obtained a very broad set of FPT algorithms, but for committee scoring rules the picture is more complex; for example, even though the Chamberlin–Courant rule has an FPT algorithm, a rule with a very similar definition (using OWA $(0, 1, 0, \dots, 0)$ instead of $(1, 0, \dots, 0)$) turned out to be $W[1]$ -hard. Yet, we have shown that FPT algorithms for Thiele rules yield FPT-approximation schemes for many committee scoring rules (in particular, those defined using Borda preferences). One of the most challenging open problems is the parameterized complexity for the Harmonic Borda rule (that is, for OWA-WINNER with Borda preferences and OWA $(1, 1/2, \dots, 1/k)$).

7 Acknowledgments

We are very thankful to anonymous reviewers of AAI for their insightful comments. AK was supported by the DFG project AFFA (BR 5207/1 and NI 369/15). Main work done while DK was with TU Berlin, supported by the DFG project MaMu (NI 369/19). PF was supported by the Friedrich Wilhelm Bessel Award from the Alexander von Humboldt Foundation (during his stay in Berlin) and by the NCN project 2016/21/B/ST6/01509.

References

Aziz, H.; Gaspers, S.; Gudmundsson, J.; Mackenzie, S.; Mattei, N.; and Walsh, T. 2015. Computational aspects of multi-winner approval voting. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, 107–115.

Betzler, N.; Slinko, A.; and Uhlmann, J. 2013. On the computation of fully proportional representation. *Journal of Art. Int. Research* 47:475–519.

Bredereck, R.; Faliszewski, P.; Niedermeier, R.; Skowron, P.; and Talmon, N. 2017. Mixed integer programming with convex/concave constraints: Fixed-parameter tractability and applications to multicovering and voting. *CoRR* abs/1709.02850.

Chamberlin, B., and Courant, P. 1983. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review* 77(3):718–733.

Elkind, E.; Faliszewski, P.; Skowron, P.; and Slinko, A. 2017. Properties of multiwinner voting rules. *Social Choice and Welfare* 48(3):599–632.

Erdős, P., and Turán, P. 1941. On a problem of Sidon in additive number theory, and on some related problems. *Journal of the London Mathematical Society* s1-16(4):212–215.

Faliszewski, P.; Skowron, P.; Slinko, A.; and Talmon, N. 2017a. Multiwinner rules on paths from k -Borda to Chamberlin–Courant. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 192–198.

Faliszewski, P.; Skowron, P.; Slinko, A.; and Talmon, N. 2017b. Multiwinner voting: A new challenge for social choice theory. In Endriss, U., ed., *Trends in Computational Social Choice*. AI Access Foundation.

Faliszewski, P.; Lackner, M.; Peters, D.; and Talmon, N. 2018a. Effective heuristics for committee scoring rules. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 1023–1030.

Faliszewski, P.; Skowron, P.; Slinko, A.; and Talmon, N. 2018b. Multiwinner analogues of the plurality rule: axiomatic and algorithmic perspectives. *Social Choice and Welfare* 51(3):513–550.

Faliszewski, P.; Slinko, A.; Stahl, K.; and Talmon, N. 2018c. Achieving fully proportional representation by clustering voters. *Journal of Heuristics* 24(5):725–756.

Faliszewski, P.; Skowron, P.; Slinko, A.; and Talmon, N. 2019. Committee scoring rules: Axiomatic characterization and hierarchy. *ACM Transactions on Economics and Computation* 7(1):3:1–3:39.

Kilgour, M. 2010. Approval balloting for multi-winner elections. In *Handbook on Approval Voting*. Springer. Chapter 6.

Lackner, M., and Skowron, P. 2018. Consistent approval-based multi-winner rules. In *Proceedings of the 19th ACM Conference on Electronic Commerce*, 47–48.

Lenstra, Jr., H. 1983. Integer programming with a fixed number of variables. *Mathematics of Operations Research* 8(4):538–548.

Lokshtanov, D. 2015. Parameterized integer quadratic programming: Variables and coefficients. *CoRR* abs/1511.00310.

Lu, T., and Boutilier, C. 2011. Budgeted social choice: From consensus to personalized decision making. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 280–286.

Peters, D. 2018. Single-peakedness and total unimodularity: New polynomial-time algorithms for multi-winner elections. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 1169–1176.

Procaccia, A.; Rosenschein, J.; and Zohar, A. 2008. On the complexity of achieving proportional representation. *Social Choice and Welfare* 30(3):353–362.

Skowron, P.; Yu, L.; Faliszewski, P.; and Elkind, E. 2015. The complexity of fully proportional representation for single-crossing electorates. *Theoretical Computer Science* 569:43–57.

Skowron, P.; Faliszewski, P.; and Lang, J. 2016. Finding a collective set of items: From proportional multirepresentation to group recommendation. *Artificial Intelligence* 241:191–216.

Skowron, P.; Faliszewski, P.; and Slinko, A. 2015. Achieving fully proportional representation: Approximability result. *Artificial Intelligence* 222:67–103.

Thiele, T. N. 1895. Om flerfoldsvalg. In *Oversigt over det Kongelige Danske Videnskabernes Selskabs Forhandling*. 415–441.

Yager, R. 1988. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics* 18(1):183–190.