Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)

Voting-Based Group Formation

Piotr Faliszewski
AGH University
Krakow, Poland
faliszew@agh.edu.pl

Arkadii Slinko
University of Auckland
Auckland, New Zealand
a.slinko@auckland.ac.nz

Nimrod Talmon
Weizmann Institute
Rehovot, Israel
nimrodtalmon77@gmail.com

Abstract

We study a combinatorial problem formulated in terms of the following group-formation scenario. Given some agents, where each agent has preferences over the set of potential group leaders, the task is to partition the agents into groups and assign a group leader to each of them, so that the group leaders have as high support as possible from the groups they are assigned to lead. We model this scenario as a voting problem, where the goal is to partition a set of voters into a prescribed number of groups so that each group elects its leader, i.e., their leader is a unique winner in the corresponding election. We study the computational complexity of this problem (and several of its variants) for Approval elections.

1 Introduction

Consider a situation where some agents must be split into k groups, and each group must be assigned a group leader. The agents have preferences over the set of potential group leaders and, intuitively, we would like each leader to have as high support as possible. However, since the requirement that *every* leader should have *as high support as possible* is not well-defined, initially we only insist that the group leader of each group is a unique winner under some given voting rule. If this voting rule is accepted by all agents as a measure of fairness, then no group will be tempted to change its leader. (Later we also discuss more demanding requirements for the leaders, such as having sufficient advantage over the next contender.) We are interested in the complexity of this problem.

Our scenario is deeply related to some problems of proportional representation in politics. Suppose the society must choose k representatives (among a number of candidates standing in the election), say, a parliament. It is known that splitting the electorate into districts, arranged by the territorial principle, and electing a representative in each district, leads to low-quality representation, especially in terms of representing minorities [Rae, 1967].

Alternatively, the society may consider forming districts not on the basis of geography, but on the basis of voter preferences, putting like-minded voters into the same virtual district. Then, each group can pick its representative using a voting rule. This important idea was suggested by Charles Dodgson [1884], known also as Lewis Carroll, and considered in a different form by Black [1958]. This approach was further developed by Chamberlin and Courant [1983] and by Monroe [1995]; the two rules suggested by them assign voters to representatives in such a way that each group of voters, represented by the same representative, creates a virtual district for which their representative is the Borda winner. The main difference between the rules of Monroe and of Chamberlin and Courant is that the former requires the district sizes to be as close to each other as possible, while the latter does not. Our work is very close in spirit to that of Chamberlin and Courant (see the discussion regarding related work, and later in the paper). However, we prefer to consider our results in terms of group formation since this framework is broader.

Our Results. We focus on the variant of our group-formation problem for approval-based voting rules (i.e., for rules where each agent indicates which leaders he approves of, and where the elected winner within a group is the candidate approved by the largest number of agents), mostly for the case of partitioning the agents into two groups (this setting is already challenging). Our main findings are as follows:

1. If either the number of agents or the number of potential group leaders is small, then the problem can be solved efficiently (formally speaking, the problem is in FPT for the parameterizations by the number of agents or the number of potential group leaders).
2. For the case of partitioning the agents into two groups, the problem is in P if either each agent approves at most two group leaders, or each agent approves almost all group leaders with the exception of one or two.
3. The problem becomes NP-hard as soon as each agent approves up to six group leaders (even for two groups).

In addition to the basic setting, we consider three variants of the problem. In the first one, we require the selected leaders to be approved by sufficiently many more agents than their next-best contenders. In the second one, we are given the sizes of the groups that we should form (quite a natural requirement in many applications). In the third one, the group leaders are part of the input and the goal is to check if there is a partition where each leader is a unique winner in some group (for this case, our NP-hardness holds already for the case where each agent approves up to three group leaders).

Related Work. Team formation and group formation are among the main research topics in the field of artificial intelligence and multiagent systems, and has been analyzed from many different perspectives. In a number of works, researchers take the view that a group of interdependent agents needs or wants to find teams that would satisfy their goals (see, e.g., the works of Shehory and Kraus [1998], Dignum et al. [2000], Gaston and desJardins [2005], Anagnostopoulos et al. [2012], and Bredereck et al. [2013]). In another line of work, the team-formation problem is viewed as a coalition formation task, i.e., through the lens of cooperative game theory (see the works of Sandholm et al. [1999] or Rahwan et al. [2012; 2011] as examples). Our case, with the centralized group (or district) formation (by a manager or an electoral designer) is, however, much similar to voting scenarios studied in computational social choice literature, specifically (1) winner determination for Chamberlin–Courant and Monroe elections, and (2) control by partition of voters [Bartholdi *et al.*, 1992] (however, this should not be confused with works on teams of voting agents such as those of Marcolino et al [2013; 2015]). We discuss these two connections now.

Under the approval-based Chamberlin–Courant rule (see the original work of Chamberlin and Courant [1983] as well as the papers of Procaccia et al. [2008], Betzler et al. [2013], Skowron and Faliszewski [2015], and Aziz et al. [2015] for a more general view of similar voting rules), the goal is to pick a committee of k candidates that, in some sense, best represents the society of voters. Formally, a given size-k committee receives one point from each voter that approves at least one of the committee members, and the committee with the most points wins. This is similar to our setting, where the set of assigned group leaders can be viewed as such a committee. However, the problems are different. For example, choosing a constant-sized committee is in P under the approval-based Chamberlin–Courant rule [Betzler *et al.*, 2013], but our problem—even for the case of partitioning agents into two groups—is NP-hard.

The problem of (constructive) control by partition of voters was introduced by Bartholdi et al. [1992] and has been studied in detail since (most notably by Hemaspaandra et al. [2007], but see also the survey of Faliszewski and Rothe [2015]). In it, we are given the voters' preferences and ask if it is possible to partition the voters (into two groups) so that a given distinguished candidate wins in the following two-round election: First, the two groups elect winners using a prescribed single-winner rule (independently from each other) and then the two selected candidates stand in the final election (under the same rule) where all the voters vote.

For the Approval rule, Hemaspaandra et al. [2007] show that this problem is NP-hard. Our problem is similar (especially to the ties-eliminate variant of the control problem, where a candidate has to be a unique winner in her subelection to pass to the next round), but in our problem it does not matter how these winners compare to each other; as a result, also our proofs work very differently compared to those of Hemaspaandra et al. [2007]. Indeed, our task seems simpler, therefore it is somewhat disappointing that we still face NP-hardness.

Finally, our work is related to papers concerning the influence of tie-breaking procedures on the complexity of strategic voting [Obraztsova and Elkind, 2011; Obraztsova *et al.*, 2011].

2 Preliminaries

We formulate our group-formation problems in the language of elections and in this section we provide the necessary definitions. We assume familiarity with standard notions regarding algorithms and complexity theory. For a positive integer z, we write $[z]$ to denote the set $\{1, \dots, z\}$.

Elections. An election $E = (C, V)$ consists of a set C of m candidates and a collection V of n voters $v_1, v_2, \dots, v_n$. We do not assume that the voters and candidates are necessarily distinct, i.e., voters may elect the leaders among their ranks. For convenience, we usually refer to the voters as males and to the candidates as females. Each voter is associated with his preference over the candidates; throughout this paper we assume that the preferences are defined in the form of sets of approved candidates. That is, each voter has a set of candidates that he approves of (and he disapproves of—sometimes we say vetoes—the other ones). A voter which approves a candidate c is said to be a c-voter.

A winner of such an election, called an approval winner, is a candidate that is approved by the highest number of voters. It is possible that several candidates get the same maximal number of approvals and, then, we say that they all tie as winners. Usually some kind of tie-breaking must be employed; however, our goal would be to achieve unique clear winners in each group, such that the tie-breaking is not required for their determination.

There are several special cases of this setting, depending on how many candidates each voter is asked to approve of:

***t*-Approval and *t*-Veto (and Plurality and Veto).** If each voter has to approve exactly t candidates (respectively, has to approve all but t candidates) then we say that the election is held according to the t-Approval rule (respectively, the t-Veto rule). 1-Approval is known as the Plurality rule, while 1-Veto is known as the Veto rule. (For t-Veto, it is sometimes more convenient to count not approvals but the number of vetoes, i.e., the number of times each candidate is not approved. The number of vetoes of a candidate is her Veto score.)

Approval. If each voter can approve any number of candidates, then we say that the election is held according to the Approval voting rule (note that all our NP-hardness results for t-Approval immediately translate to NP-hardness results for Approval).

For a voting rule $\mathcal{R}$ (in our case, one of t-Approval/t-Veto/Approval) and an election $E = (C, V)$, we denote the set of winners of E under $\mathcal{R}$ by $\mathcal{R}(C, V)$.

The Problem. We study the computational complexity of the following problem.

$\mathcal{R}$-k-PARTITION
Input: An election $E = (C, V)$ and a positive integer k.
Question: Is there a partition of V into subsets $V_1, \ldots, V_k$ such that $|\mathcal{R}(C, V_i)| = 1$ for each $i \in [k]$, and $\mathcal{R}(C, V_i) \neq \mathcal{R}(C, V_j)$ whenever $i \neq j$?

That is, the task is to partition the voters into k groups (each voter should belong to exactly one group) so that in each group there is a unique $\mathcal{R}$-winner, distinct from all others. $\mathcal{R}$-k-PARTITION is the vanilla variant of our problem. We also consider the following variants (we describe each of them by specifying how it differs from $\mathcal{R}$-k-PARTITION):

1. In the $\mathcal{R}$-MAXGAP-k-PARTITION variant, in addition to the standard parts of the input, we are also given a number t, and ask whether there is a partition of V into $V_1, \ldots, V_k$ so that, for each i, the unique winner in V_i has at least t points more than the next-best candidate(s).
2. In the $\mathcal{R}$-SIZED-k-PARTITION variant, in addition to the standard parts of the input, we are also given k numbers, $s_1, \ldots, s_k$, and ask whether there is a partition of V into $V_1, \ldots, V_k$ such that, for each i, V_i has a unique winner and $|V_i| = s_i$.
3. In the $\mathcal{R}$-PRESELECTED-k-PARTITION variant, in addition to the standard parts of the input, we are also given the desired k group leaders, and the task is to check whether there is a partition of the voters so that each group leader is a unique winner for exactly one group.

The $\mathcal{R}$-PRESELECTED-k-PARTITION variant is interesting for two reasons. First, we might be in the setting where the designer has already identified the set of desirable group leaders and would like to check if it is possible to form groups that give the preselected leaders necessary support. Second, most of our results translate from this case to the other scenarios.

For the other two problems, $\mathcal{R}$-MAXGAP-k-PARTITION speaks of situations where we maximize the (relative) support of the selected group leaders, and $\mathcal{R}$-SIZED-k-PARTITION deals with the case where groups have to be of specified sizes.

Comparison to Chamberlin–Courant. Our problems are similar to, but different from, that of winner determination under Approval-based Chamberlin–Courant rule ($\mathcal{R}$-k-CC):

$\mathcal{R}$-k-CC
Input: An election $E = (C, V)$ and a positive integer k.
Task: Find a subset $\{c_1, \ldots, c_k\}$ of candidates and a partition of the voters into k groups, $V_1, \ldots, V_k$, so that the sum of approval scores of c_1 in V_1, c_2 in V_2, …, c_k in V_k is maximal.

We mention that $\mathcal{R}$-k-CC is in P for the cases where k is fixed [Betzler *et al.*, 2013], whereas we will see that already 6-APPROVAL-2-PARTITION is NP-hard.

Parameterized Complexity. An instance (I, k) of a parameterized problem consists of an instance I of a non-parameterized problem and an integer k, referred to as the *parameter* (see, e.g., the textbook of Niedermeier [2006] for more details). A parameterized problem is called *fixed-parameter tractable* (or, is said to be in FPT) if there is an algorithm solving it in $f(k) \cdot |I|^{O(1)}$ time, where f is some computable function. Betzler et al. [2012] survey parameterized complexity results in voting.

3 Results

We first present the results for the case of preselected group leaders. Then, based on those, we give results for all the other scenarios. Finally, we consider the case where we have either only a few candidates or only a few voters. We mostly focus on the case of partitioning the voters into two groups, but occasionally (e.g., in the next two theorems) we prove results for general values of k.

3.1 Results for Preselected Group Leaders

Plurality is a simple and fundamental voting rule, so it is not surprising that we get a polynomial-time algorithm for it.

Theorem 1. PLURALITY-PRESELECTED-k-PARTITION *is in* P.

Proof. Consider an instance of Plurality-Preselected-k-Partition, where the preselected leaders are $p_1, \ldots, p_k$. If for some $i \in [k]$ there is no voter approving p_i, then we obviously reject. If we did not reject at the first stage, we check if $n - k \geq 2Q$, where Q is the number of voters who approve none of the preselected leaders. We accept if this is true and reject otherwise. To justify this, for each $i \in [k]$, let P_i be the number of voters who approve p_i. For each i, the ith group of voters in the partition can accommodate the P_i voters that approve of p_i and up to $P_i - 1$ voters who do not approve of p_i. Then, we have a yes-instance iff $\sum_{i=1}^{k}(P_i - 1) \geq Q$. Since $\sum_{i=1}^{k}(P_i - 1) = n - Q - k$, we have a yes-instance if and only if $n - k \geq 2Q$. □

For the case of Veto, we need a bit more involved reasoning. Due to space constraints, we omit the corresponding proof, as well as some other later proofs. From now on we will not mention this explicitly.

Theorem 2. VETO-PRESELECTED-k-PARTITION *is in* P.

The fact that we can quite easily deal with Plurality (and Veto) is encouraging. Unfortunately, going beyond these two simplest rules becomes more challenging. We still obtain polynomial-time algorithms for 2-Approval and 2-Veto, but only for the case of partitioning the voters into two groups.

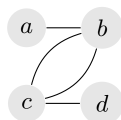
Theorem 3. 2-APPROVAL-PRESELECTED-2-PARTITION *is in* P.

Proof. Given an instance of 2-Approval-Preselected-2-Partition that includes an election $E = (C, V)$ and two preselected leaders, p_1 and p_2, we will show how to check if it is possible to partition V into V_1 and V_2, so that p_1 is a unique winner in V_1 and p_2 is a unique winner in V_2.

It is clear that we can assign all the voters who approve p_1 but not p_2 to V_1, and all the voters who approve p_2 but not p_1 to V_2. Regarding voters who approve both p_1 and p_2, we proceed as follows: Let q be the number of such voters. We guess a number $q' \leq q$ and assign q' of these voters to V_1, while the remaining ones we assign to V_2 (clearly, it is irrelevant which specific voters we assign to each of the groups). Let U_1 and U_2 denote the voters already allocated to be in V_1 and V_2.

We might have some remaining voters who approve neither p_1 nor p_2. We will assign parts for these voters by first translating the problem to a multigraph problem. We form a vertex for each candidate in $C \setminus \{p_1, p_2\}$. For each not-yet-allocated voter v (i.e., for each voter from $V \setminus (U_1 \cup U_2)$), we form an edge between the two vertices that correspond to the candidates approved by v. This gives us a multigraph which we denote as G. Consider the following illustrating example.

Example 1. Consider an election with candidates $\{a, b, c, d, p_1, p_2\}$ and the situation after we have already allocated the voters who approve of p_1 and/or p_2. Imagine we have four unallocated voters, such that the first one approves a and b, the second and the third one both approve b and c, and the fourth one approves c and d. We then construct the following multigraph.



The score that each candidate receives from the unallocated voters is equal to the degree of the corresponding vertex. △

The problem of assigning the unallocated voters to either V_1 or V_2 without meeting-or-exceeding the scores of p_1 and p_2, respectively, can now be reduced to the (g, f)-FACTOR problem defined as follows (for a multigraph G and some vertex v in it, we write $\deg_G(v)$ to denote its degree). By $\mathbb{N}_0$ we denote the set of all non-negative integers.

(g, f)-FACTOR
Input: A multigraph $G = (V, E)$ and two functions $g, f \colon V \to \mathbb{N}_0$, with $g(v) \le f(v)$ for each $v \in V$.
Question: Does G contain an (g, f)-*Factor*, that is, a subgraph $G_0 = (V, E_0)$ of G such that $g(v) \le \deg_{G_0}(v) \le f(v)$ for all $v \in V$?

It is known that (g, f)-FACTOR is polynomial-time solvable [Anstee, 1985] (see also the prior work of Gabow [1983]).

We use the following notation to describe the specific instance of (g, f)-FACTOR that we form. Recall that G is the multigraph we have constructed. Note that, for each vertex c in G, its degree $\deg_G(c)$ is equal to the Approval score of c in the original election where we remove the voters already assigned to V_1 and V_2. For each vertex c and each $i \in \{1, 2\}$, let us denote the number of voters who are assigned to part V_i and approve c by $score_{U_i}(c)$ (this is the 2-approval score of c in the subelection U_i).

We note that $score_{U_i}(p_i) - score_{U_i}(c) - 1$ is the number of not-yet-allocated voters which approve c, that can be added to U_i without preventing p_i from being the 2-approval winner in the resulting subelection. We are now ready to describe the instance of (g, f)-FACTOR that we solve. For each vertex c of graph G we define the following two values:

$$g(c) := \deg_G(c) - (score_{U_2}(p_2) - score_{U_2}(c) - 1), \quad (1)$$
$$f(c) := score_{U_1}(p_1) - score_{U_1}(c) - 1. \quad (2)$$

If for some c it holds that $g(c) > f(c)$, then our guess of the number q' was unsuccessful. Otherwise, if $g(c) \le f(c)$ for all vertices of G, then, using a polynomial-time algorithm for (g, f)-FACTOR, we compute the graph G_0. If it does not exist, then we reject (this guess of q'). If it exists then we obtain the desired partition of voters, and we assign all the voters corresponding to the edges in G_0 to part V_1, and all the remaining voters to part V_2. Note that, for each candidate c, it is not possible to assign more than $f(c)$ edges to V_1, since otherwise c would win in part V_1. Similarly, at least $g(c)$ edges must be assigned to V_1, since otherwise c would win in part V_2. Thus, the algorithm is correct (i.e., if there is a solution for the input instance of 2-APPROVAL-PRESELECTED-2-PARTITION then the algorithm accepts for some guessed q', and otherwise it rejects for all q'). □

We obtain a similar result for the case of the 2-Veto rule. While the proof also relies on using the (g, f)-FACTOR problem, it requires some additional tricks.

Theorem 4. 2-VETO-PRESELECTED-2-PARTITION *is in* P.

There are two ways of extending the above two theorems. We can consider either t-{APPROVAL, VETO}-PRESELECTED-2-PARTITION for $t > 2$, or 2-{APPROVAL, VETO}-PRESELECTED-k-PARTITION for $k > 2$. For the former cases, we show NP-hardness (we omit the proofs here, but later we describe how to adapt the proof of Theorem 8 to apply to Theorem 5).

Theorem 5. 3-APPROVAL-PRESELECTED-2-PARTITION *is* NP-*hard.*

Theorem 6. 3-VETO-PRESELECTED-2-PARTITION *is* NP-*hard.*

The following two corollaries are achieved by simple adaptations of the reductions used in the proofs of Theorem 5 and Theorem 6.

Corollary 1. t-APPROVAL-PRESELECTED-k-PARTITION *is* NP-*hard whenever* $t \ge 3$ *and* $k \ge 2$.

Corollary 2. t-VETO-PRESELECTED-k-PARTITION *is* NP-*hard whenever* $t \ge 3$ *and* $k \ge 2$.

The other line of extending the results for 2-{APPROVAL, VETO}-PRESELECTED-2-PARTITION is more problematic. As far as polynomial-time algorithms are concerned, the simplest case to consider would be that of 2-APPROVAL-PRESELECTED-3-PARTITION. It appears that the ideas we used for our polynomial-time algorithms cannot be generalized to this case: We would need to solve a problem which has a lot of similarities with the NP-hard EDGE COLORING problem [Garey and Johnson, 1979]. Yet, we could not prove NP-hardness for any variant of the 2-APPROVAL-k-PARTITION problem, for $k \ge 3$; the complexity of these cases remains open.

3.2 Other Variants of the Problem

Most of our results for preselected leaders still hold for the other variants of the problem, with only a few exceptions.

First, it holds that PLURALITY-k-PARTITION and VETO-k-PARTITION are in P. To show this, we use algorithms that are somewhat different from those used in the case of preselected leaders. For the case of creating two groups, we show that, whenever we have a polynomial-time algorithm for $\mathcal{R}$-PRESELECTED-2-PARTITION, we automatically get an algorithm for $\mathcal{R}$-2-PARTITION.

Proposition 1. *Let $\mathcal{R}$ be a voting rule. If $\mathcal{R}$-*PRESELECTED-2-PARTITION *is in* P *then so is $\mathcal{R}$-*2-PARTITION.

Proof. Consider an instance of $\mathcal{R}$-2-PARTITION, with candidate set C. It suffices to run the polynomial-time algorithm for $\mathcal{R}$-PRESELECTED-2-PARTITION for each possible choice of two distinct group leaders. If the algorithm accepts for any such choice, then we accept; otherwise, we reject. □

Naturally, similar approach works for more groups, provided that their number is upper bounded by a constant, but for our results the above proposition is sufficient. Altogether, we get the following.

Theorem 7. PLURALITY-k-PARTITION, VETO-k-PARTITION, 2-APPROVAL-2-PARTITION, *and* 2-VETO-2-PARTITION *are in* P.

Unfortunately, we were not able to show NP-hardness of t-APPROVAL-2-PARTITION for $t \in \{3, 4, 5\}$, but we do get NP-hardness for the 6-Approval case.

Theorem 8. 6-APPROVAL-2-PARTITION *is* NP*-hard.*

Proof. We provide a reduction from the following NP-hard problem [Gonzalez, 1985].

RESTRICTED EXACT COVER BY 3-SETS
Input: Set of elements $X = \{x_1, \dots, x_n\}$, collection $\mathcal{S} = \{S_1, \dots, S_n\}$ of size-3 subsets of X such that each x_i appears in exactly three sets.
Question: Is there a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ of sets such that each element x_i appears exactly once in $\mathcal{S}'$?

Given an instance for RESTRICTED EXACT COVER BY 3-SETS, we create an instance for 6-APPROVAL-2-PARTITION as follows. For each element x_i, we create two candidates, x_i and x'_i. We also create two special candidates, a and b. For each set S_i, we create a voter S_i who approves the set of candidates $\{x_i, x'_i : x_i \in S_i\}$. Moreover, we create two voters who approve a and three voters who approve b. This finishes the reduction, which is computable in polynomial time.[1]

We omit the full proof of correctness, but only mention that, given a solution for RESTRICTED EXACT COVER BY 3-SETS, that is, a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that each element appears exactly once in $\mathcal{S}'$, we partition the voters as follows. We assign to part V_1 all the voters that correspond to sets from $\mathcal{S}'$, as well as the two voters that approve a; we assign all the other voters to part V_2. Since each element x_i appears exactly once in $\mathcal{S}'$, it follows that each candidate x_i and x'_i appears exactly once in V_1. □

[1]Note that the reduction, as stated above, generate an Approval but not 6-Approval elections as, in the latter, voters who approve a or b should also approve some five other candidates. This can be easily fixed by adding 25 dummy candidates.

Remark 1. The above proof also works for Theorem 5: we fix a and b as the preselected leaders and remove $x'_1, \dots, x'_n$.

Let us consider the MAXGAP variant. It turns out we can get the following results, by using techniques similar to those we have used so far. For example, to show that 2-APPROVAL-MAXGAP-2-PARTITION is in P, we take the same approach as in Theorem 3: we iterate over all choices of group leaders (as in Theorem 1) and in equations (1) and (2) we replace "-1" with "$-t$," where t is the minimum score advantage any group leader must have over the next-best contender.

Theorem 9. PLURALITY-MAXGAP-k-PARTITION, VETO-MAXGAP-k-PARTITION, 2-APPROVAL-MAXGAP-2-PARTITION, *and* 2-VETO-MAXGAP-2-PARTITION *are in* P *while* 6-APPROVAL-MAXGAP-2-PARTITION *and* 6-VETO-MAXGAP-2-PARTITION *are* NP*-hard.*

The situation is more complex for the SIZED variant. For example, we did not obtain results for PLURALITY-SIZED-k-PARTITION for the case where k is part of the input. Indeed, our algorithm for PLURALITY-SIZED-2-PARTITION is quite different from all other ones, thus, for this case, we provide a proof after the theorem statement.

Theorem 10. PLURALITY-SIZED-2-PARTITION, VETO-2-SIZED-2-PARTITION, 2-APPROVAL-SIZED-2-PARTITION, *and* 2-VETO-SIZED-2-PARTITION *are in* P *while* 6-APPROVAL-SIZED-2-PARTITION *and* 6-VETO-SIZED-2-PARTITION *are* NP*-hard.*

Proof for PLURALITY-SIZED-2-PARTITION. We begin by guessing a pair of candidates, a_1 and a_2, which will be the winners of V_1 and V_2, respectively.

Let T_1 be the number of a_1-voters in the election and T_2 be the number of a_2-voters in the election (recall that a c-voter is a voter who approves c). Due to the given size constraints, i.e., that $|V_1| = s_1$ and $|V_2| = s_2$, we cannot simply assign all a_1-voters to V_1 and all a_2-voters to V_2. Instead, we begin by guessing a value Z_1, which is the number of a_1-voters that we assign to V_1, as well as a value Z_2, which is the number of a_2-voters that we assign to V_2. After fixing these values and assigning the relevant voters according to them, the problem can be reduced to finding a maximum flow in the network constructed as follows (note that at this point we assigned $Z_1 + (T_2 - Z_2)$ voters to V_1 and $Z_2 + (T_1 - Z_1)$ voters to V_2; if this already breaks the size constraints, or prevents a_1 from winning in V_1, or prevents a_2 from winning in V_2, then we try a different set of guesses).

We construct a source node s and a target node t. For each candidate $c \notin \{a_1, a_2\}$, we construct a node v_c (recall that all the a_1-voters and all the a_2-voters are already assigned to groups). We construct two additional nodes, t_1 (corresponding to V_1) and t_2 (corresponding for V_2). For each v_c, we create an arc (s, v_c) with capacity equal to the number of c-voters in the election. Further, for each v_c, we create an arc (v_c, t_1) with capacity $Z_1 - 1$, as well as an arc (v_c, t_2) with capacity $Z_2 - 1$. We create an arc (t_1, t) with capacity $s_1 - Z_1 - (T_2 - Z_2)$, as well as an arc (t_2, t) with capacity $s_2 - Z_2 - (T_1 - Z_1)$. This finishes the description of the network. A maximum flow in this network can be found in polynomial-time. If the network can be completely saturated (i.e., all the edges from s can be used to their maximum capacity), then we accept; otherwise, we reject. Further, for each $c \notin \{a_1, a_2\}$, the amount of flow on the arc (v_c, t_1) (respectively, (v_c, t_2)) corresponds to the number of c-voters which will be assigned to V_1 (respectively, V_2).

The correctness of the algorithm follows by noticing that, for correct guesses of the values Z_1 and Z_2, and for each candidate $c \notin \{a_1, a_2\}$, no more than $Z_1 - 1$ (respectively, $Z_2 - 1$) c-voters can be assigned to V_1 (respectively, V_2). Further, the capacities on the arcs (t_1, t) and (t_2, t) make sure that, when the network is saturated, the sizes of the groups will be as required. □

3.3 Few Candidates or Few Voters

We conclude the technical discussion by showing that the $\mathcal{R}$-k-PARTITION problem is fixed-parameter tractable with respect to parameterizations either by the number n of voters or by the number m of candidates. Note that these algorithms work not only for approval-based voting rules, but for a much larger family of voting rules, including, for example, all scoring rules.

Proposition 2. *Let $\mathcal{R}$ be a voting rule for which the winner-determination problem is fixed-parameter tractable with respect to the number n of voters. Then, $\mathcal{R}$-k-PARTITION is also fixed-parameter tractable with respect to the number n of voters.*

By formulating the $\mathcal{R}$-k-PARTITION problem as a linear integer program (ILP) and applying a famous result of Lenstra [1983], we get the next result, for all consistent voting rules (including all the rules studied in this paper).

Definition 1. A voting rule is *consistent* if, for every pair of elections, $E_1 = (C, V_1)$ and $E_2 = (C, V_2)$, it holds that if $\mathcal{R}(C, V_1) \cap \mathcal{R}(C, V_2) \neq \emptyset$ then $\mathcal{R}(C, V_1 + V_2) = \mathcal{R}(C, V_1) \cap \mathcal{R}(C, V_2)$.

Proposition 3. *Let $\mathcal{R}$ be a consistent voting rule for which winner-determination can be decided by an integer linear program where the number of variables is upper-bounded by a function which depends only on the number m of candidates. Then, $\mathcal{R}$-k-PARTITION is fixed-parameter tractable with respect to the number m of candidates.*

These results mean that, after all, the NP-hardness results from the preceding sections are not necessarily problematic if we have only a few voters or only a few candidates.

4 Conclusions and Future Research

We have shown that the problem of partitioning a group of agents into groups that have clear leaders is surprisingly difficult. It is NP-hard already for two groups, for the case where each agent approves at most six possible leaders (and if we want to have a predetermined set of leaders for the groups, then it is NP-hard already when each agent approves at most three possible leaders). Nonetheless, we have also found a number of cases where our problem can be solved efficiently.

Our hardness results are somewhat unexpected, especially when contrasted with the easiness of the winner-determination problem under the approval-based Chamberlin–Courant rule. For example, the problem of finding a winning committee of size two under this rule is in P, but we show that finding such a committee and partitioning the voters into two groups, each represented by their assigned committee member, is NP-hard if we insist that the committee members are the unique most-supported candidates among the voters they represent.

	PRESELECTED-k-PARTITION		
	$k = 2$	$k > 2$	Reference
Plurality	P	P	Theorem 1
Veto	P	P	Theorem 2
2-Approval	P	Open	Theorem 3
2-Veto	P	Open	Theorem 4
$\geq$ 3-Approval	NP-hard	NP-hard	Corollary 1
$\geq$ 3-Veto	NP-hard	NP-hard	Corollary 2

Table 1: Summary of our results for the variant with preselected leaders. For the non-preselected variant, we have NP-hardness only for 6-Approval and 6-Veto. The results hold also for the two generalizations considered in this paper, namely, where we (1) require a specific score advantage for the leaders, or (2) impose constraints on the group sizes.

Our results lead to a number of avenues for future research. To mention just a few, it would be natural to study ordinal voting rules (where the voters rank the candidates instead of giving approvals) and it would be interesting to study restricted domains (e.g., variants of single-peakedness and single-crossingness for Approval voting (see the works of Faliszewski et al. [2011] and Elkind and Lackner [2015]).

Acknowledgments. Piotr Faliszewski was supported NCN grant DEC-2012/06/M/ST1/00358. Arkadii Slinko was supported by the Royal Society of NZ Marsden Fund UOA-254. Nimrod Talmon was supported by a postdoctoral fellowship from I-CORE ALGO.

References

[Anagnostopoulos *et al.*, 2012] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi. Online team formation in social networks. In *Proceedings of the 21st International Conference on World Wide Web (WWW'12)*, pages 839–848. ACM Press, April 2012.

[Anstee, 1985] R. Anstee. An algorithmic proof of Tutte's f-factor theorem. *Journal of Algorithms*, 6(1):112–131, 1985.

[Aziz *et al.*, 2015] H. Aziz, M. Brill, V. Conitzer, E. Elkind, R. Freeman, and T. Walsh. Justified representation in approval-based committee voting. In *Proceedings of AAAI-2015*, pages 784–790, 2015.

[Bartholdi *et al.*, 1992] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[Betzler *et al.*, 2012] N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier. Studies in computational aspects of voting—a parameterized complexity perspective. In *The Multivariate Algorithmic Revolution and Beyond*, volume 7370 of *LNCS*, pages 318–363. Springer, 2012.

[Betzler *et al.*, 2013] N. Betzler, A. Slinko, and J. Uhlmann. On the computation of fully proportional representation. *Journal of Artificial Intelligence Research*, 47:475–519, 2013.

[Black, 1958] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.

[Bredereck *et al.*, 2013] R. Bredereck, T. Köhler, A. Nichterlein, R. Niedermeier, and G. Philip. Using patterns to form homogeneous teams. *Algorithmica*, 71(2):517–538, 2013.

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

[Dignum *et al.*, 2000] F. Dignum, B. Dunin-Keplicz, and R. Verbrugge. Agent theory for team formation by dialogue. In *Proceedings of the 7th International Workshop on Agent Theories Architectures and Language (ATAL'00)*, pages 150–166. Springer-Verlag, July 2000.

[Dodgson, 1884] C. Dodgson. *The Principles of Parliamentary Representation*. Harrison, London, 1884.

[Elkind and Lackner, 2015] E. Elkind and M. Lackner. Structure in dichotomous preferences. In *Proceedings of IJCAI-15*, pages 2019–2025, July 2015.

[Faliszewski and Rothe, 2015] P. Faliszewski and J. Rothe. Control and bribery in voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 7. Cambridge University Press, 2015.

[Faliszewski *et al.*, 2011] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Information and Computation*, 209(2):89–107, 2011.

[Gabow, 1983] H. N. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems (stoc'83). In *Proceedings of the 15th ACM Symposium on Theory of Computing*, pages 448–456. ACM, 1983.

[Garey and Johnson, 1979] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[Gaston and desJardins, 2005] M. Gaston and M. desJardins. Agent-organized networks for dynamic team formation. In *Proceedings of AAMAS-2005*, pages 230–237. ACM Press, May 2005.

[Gonzalez, 1985] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[Hemaspaandra *et al.*, 2007] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

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

[Marcolino *et al.*, 2013] L. Marcolino, A. Xin Jiang, and M. Tambe. Multi-agent team formation: Diversity beats strength? In *Proceedings of IJCAI-13*, pages 279–285, August 2013.

[Marcolino, 2015] L. Marcolino. Unleashing the power of multi-agent voting teams. In *Proceedings of IJCAI-15*, pages 4399–4400. AAAI Press, July 2015.

[Monroe, 1995] B. Monroe. Fully proportional representation. *American Political Science Review*, 89(4):925–940, 1995.

[Niedermeier, 2006] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[Obraztsova and Elkind, 2011] S. Obraztsova and E. Elkind. On the complexity of voting manipulation under randomized tie-breaking. In *Proceedings of IJCAI-11*, pages 319–324, July 2011.

[Obraztsova *et al.*, 2011] S. Obraztsova, E. Elkind, and N. Hazon. Ties matter: Complexity of voting manipulation revisited. In *Proceedings of AAMAS-2011*, pages 71–78, May 2011.

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

[Rae, 1967] D. Rae. *The Political Consequences of Electoral Laws*. New Haven: Yale University Press, 1967.

[Rahwan *et al.*, 2011] T. Rahwan, T. Michalak, E. Elkind, P. Faliszewski, J. Sroka, M. Wooldridge, and N. Jennings. Constrained coalition formation. In *Proceedings of AAAI-2011*, pages 719–725, August 2011.

[Rahwan *et al.*, 2012] T. Rahwan, T. Michalak, M. Wooldridge, and N. Jennings. Anytime coalition structure generation in multi-agent systems with positive or negative externalities. *Artificial Intelligence*, 186:95–122, 2012.

[Sandholm *et al.*, 1999] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1–2):209–238, 1999.

[Shehory and Kraus, 1998] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1–2):165–200, 1998.

[Skowron and Faliszewski, 2015] P. Skowron and P. Faliszewski. Fully proportional representation with approval ballots: Approximating the MaxCover problem with bounded frequencies in FPT time. In *Proceedings of AAAI-2015*, pages 2124–2130, January 2015.