Well-Structured Committees

Sushmita Gupta[1] , **Pallavi Jain**[2] and **Saket Saurabh**[3,4]
[1]National Institute of Science Education and Research (NISER), HBNI, India
[2]Ben-Gurion University of the Negev, Israel,
[3]The Institute of Mathematical Sciences, HBNI, India,
[4]University of Bergen, Norway
sushmitagupta@niser.ac.in, pallavi@post.bgu.ac.il, and saket@imsc.res.in

Abstract

In the standard model of committee selection we are given a set of ordinal votes over a set of candidates and a desired committee size, and the task is to select a committee that relates to the given votes. Motivated by possible interactions and dependencies between candidates, we study a generalization of committee selection in which the candidates are connected via a network and the task is to select a committee that also satisfies certain properties with respect to this candidate network. To accommodate few possibilities of taking voter preferences into account, we consider three standard and diverse voting rules (namely, k-Borda, Chamberlin–Courant, and Gehrlein stability); to model different aspects of interactions and dependencies between candidates, we consider two graph properties (namely, Independent Set and Connectivity). We study the parameterized complexity of the corresponding combinatorial problems and discuss certain implications of our algorithmic results.

1 Introduction

Given a set of candidates and a collection of voters – each expressing her preferences over the candidates – a natural task is to select a committee of candidates that would correspond to the voter opinions. The study of selecting good committees is usually approached through the subfield of computational social choice [Brandt *et al.*, 2016; Endriss, 2017] known as multiwinner elections [Faliszewski *et al.*, 2017a]). This study has proven to be quite fruitful: Among other topics, researchers have studied the complexity of winner determination for various multiwinner voting rules (see, e.g., [Procaccia *et al.*, 2008; Brill *et al.*, 2019]); the complexity of related manipulative actions (see, e.g., [Procaccia *et al.*, 2007; Faliszewski *et al.*, 2017b]); few frameworks of such rules (see, e.g., [Faliszewski *et al.*, 2016; Faliszewski *et al.*, 2018]); and certain related axiomatic properties (see, e.g., [Elkind *et al.*, 2017; Aziz *et al.*, 2017a]).

One aspect that is usually overlooked is the possibility of certain interactions and dependencies between candidates. For example, some candidates might be incompatible to a certain extent, perhaps due to different personalities; or some candidates might possess more efficient means of communication among them, allowing them to cooperate more effectively than others (we explain this in detail in the paragraph "Candidate Networks and Graph Properties"). Indeed, some papers do take such externalities between candidates into account; e.g., Bredereck et al. [2018] (and others) study a setting in which candidates have attributes (e.g., occupation), and the task is to select a committee that is diverse enough. Izsak et al. [2018] study a setting of a candidate network modeling certain submodularities and supermodularities between groups of candidates. Talmon [2018] studies certain related interactions wrt a voter network (and not a candidate network).

Here we consider the following setting (formal definitions are given in the paragraph "Voter Preferences and Candidate Networks"): Given a graph over a set $\mathcal{C}$ of m candidates – a *candidate graph*; a collection of n ordinal votes over the same set $\mathcal{C}$ (that is, each of the n corresponding voters ranks all the m available candidates) – a *preference profile*; and a desired committee size k, we are interested in finding a committee that is "good" with respect to both (1) the voter preferences and (2) the candidate graph. Our setting is parameterized by both (1) a multiwinner voting rule $\mathcal{R}$, which takes the collection of n ordinal votes and assigns a score to each possible committee (a *committee* is a subset of $\mathcal{C}$ containing exactly k candidates) based on the voter preferences alone and (2) a graph property $\mathcal{Q}$, which takes the candidate graph and look for each possible committee (corresponding to a vertex subset in the graph) that satisfies property $\mathcal{Q}$ based on the candidate graph alone. For specific $\mathcal{R}$ and $\mathcal{Q}$, we are interested in designing efficient algorithms that find a committee with the highest $\mathcal{R}$-score possible among all committees satisfying $\mathcal{Q}$. We describe our specific $\mathcal{R}$'s and $\mathcal{Q}$'s further on.

While, as we show, the combinatorial problems we study here are generally intractable, we nevertheless develop several efficient algorithms for certain special cases of these problems. In particular, we discuss certain problem parameters that allow for efficient parameterized algorithms; as well as consider some domain restrictions and various restrictions on the candidate graphs that allow for efficient algorithms for these cases. As a added benefit, some of our algorithms solve known graph problems such as BUDGETED MAXIMUM WEIGHT TERMINAL STEINER TREE (BUDG-MAXWT-TST) and MAXIMUM NODE-WEIGHTED k-TREE

(MAX-NODEWT-k-TREE), thus our results are of separate interest wrt these problems.

Voter Preferences and Multiwinner Rules

In multiwinner elections the task is to select a committee that best corresponds to the voter preferences. Formally, a *multiwinner election* consists of a set $\mathcal{C}$ of m candidates, a collection of n ordinal votes (i.e., rankings) of $\mathcal{C}$, and a committee size k. A *multiwinner voting rule* takes a multiwinner election as input and outputs a set of k candidates from $\mathcal{C}$ as a *winning committee*.

Many multiwinner voting rules are studied in the literature, out of which we consider the following: Under k-Borda, a voter assigns a score of $m - i$ to a candidate she ranks at the ith position, and the winning committee consists of the k candidates with the highest sum of scores over the voters; under Chamberlin–Courant (CC), a voter assigns a score of $m - i$ to a committee out of which the candidate she ranks the highest (referred to as the voter's *representative*) is at position i, and the winning committee is of the highest score; under Gehrlein stability (GS), a committee S is chosen such that no committee member $c \in S$ has some $c' \notin S$ such that more voters rank c' higher than c than vice versa; this is known as weak Gehrlein stability in the literature [Aziz *et al.*, 2017b], we omit weak just for simplicity.

We chose k-Borda, CC, and GS as they are all well-studied and form a diverse set of multiwinner rules: In particular, following the discussion of Faliszewski et al. [Faliszewski *et al.*, 2017a], k-Borda is polynomial-time computable and an archetypal rule for selecting individually accepted candidates, CC is NP-hard and aims at identifying committees echoing all views present in society, and GS is a generalization of single-winner Condorcet-consistent rules, and it is a natural Condorcet-based multiwinner voting rule.

Candidate Networks and Graph Properties

We put voter preferences aside for a moment, and consider interactions and dependencies between candidates; consider the *candidate network*, containing one vertex c for each candidate $c \in C$ (we thus use candidates and vertices interchangeably):

- Consider a scenario of project team formation. In the election, voters are not people, but the skills required for the project. All the candidates are ranked on the basis of their skills. Now, we would like to form a team which is not only good from the perspective of skills, but also team members should not distrust and dislike each other, as this will affect the project. This can be captured by a candidate network in which there is an edge between two vertices if the corresponding candidates are "incompatible". Such compatibility should also be considered in parliamentary elections, which is usually ignored. A related situation is if there is an edge between two candidates if they are too "similar", and we wish to choose a diverse committee to perform, say, a task for which candidates with different skills are needed. In these cases, it is natural to select a committee that corresponds to an **Independent Set** of size k in the candidate network (i.e., a set of pairwise nonadjacent vertices).[1]
- Consider a scenario in which a "good" communication is required between committee members, for example, in a rescue team. Here, again candidates are ranked on the basis of their skills. This can be captured by a candidate network in which there is an edge between two vertices if there is some communication link between the corresponding candidates. Another scenario is choosing positions for routers in a network: one shall ensure router connectivity, while also taking voter preferences into consideration, corresponding to some external features of each router. To foster efficient communication, it would be natural to select a committee that is **Connected** (i.e., corresponds to a connected subgraph) in the candidate graph.

Voter Preferences and Candidate Networks.

In this paper we combine the two aspects: Voter preferences – modeled via a multiwinner voting rule and candidate networks – modeled via a graph property.

Formally, we study the following problem, parameterized by a graph property $\mathcal{Q}$ (a property that is either satisfied by a subgraph or not) and a multiwinner voting rule $\mathcal{R}$.

Definition 1. *In the* $(\mathcal{Q}, \mathcal{R})$-SCS *problem, we are given a set of m candidates, $\mathcal{C}$, a collection of n ordinal votes over $\mathcal{C}$, the committee size k, a candidate graph over $\mathcal{C}$, and the desired score s; the goal is to decide if there exists a committee satisfying $\mathcal{Q}$ whose $\mathcal{R}$-score is at least s.*

We refer to the optimization version of $(\mathcal{Q}, \mathcal{R})$-SCS, in which the goal is to find a committee with the highest score, as MAX-$(\mathcal{Q}, \mathcal{R})$-SCS. GS is slightly different, as stable committees do not always exist; for GS we require the output committee to be Gehrlein stable, as in $\mathcal{Q}$-GS, the goal is to find a stable committee also satisfying property $\mathcal{Q}$.

Let (E, H, k, s) be an instance of $(\mathcal{Q}, \mathcal{R})$-SCS, where E is a preference profile over candidates $\mathcal{C}$ and voter set $\mathcal{V}$, H is a candidate graph, k is the size of committee, and s is the score. When $\mathcal{Q}$ is the property of "independence" (i.e., we require the winning committee to form an independent set in H), we refer to $(\mathcal{Q}, \mathcal{R})$-SCS as the CONFLICT FREE-$\mathcal{R}$ (CF-$\mathcal{R}$) problem. Similarly, when $\mathcal{Q}$ is the property of "connectivity" (i.e., when we require that the winning committee be connected in H), we refer to $(\mathcal{Q}, \mathcal{R})$-SCS as the CONNECTED-$\mathcal{R}$ (CONN-$\mathcal{R}$) problem.

1.1 Initial Observations

First, we observe that $(\mathcal{Q}, \mathcal{R})$-SCS indeed generalizes winner determination wrt $\mathcal{R}$: In particular, if the candidate graph is empty, then any set of k vertices is an independent set; thus, $(\mathcal{Q}, \mathcal{R})$-SCS is reduced to finding a k-sized committee wrt $\mathcal{R}$. Similarly, if the candidate graph is a clique, then any set of k vertices is connected. (We write $\mathcal{R}$ also to refer to Winner Determination for $\mathcal{R}$).

Observation 1. *If $\mathcal{R}$ is* NP-hard *(*W-hard *wrt t), then* CF-$\mathcal{R}$ *is* NP-hard *(*W-hard *wrt t), even for empty candidate graphs; and* CONN-$\mathcal{R}$ *is* NP-hard *(*W-hard *wrt t), even when the candidate graph is a clique.*

Since CC is NP-hard and W[2]-hard wrt k [Procaccia *et al.*, 2008] and GS is NP-hard [Aziz *et al.*, 2017c] and W[1]-hard wrt k [Gupta *et al.*, 2019], using Observation 1 and the fact that $(\mathcal{Q}, \mathcal{R})$-SCS is decidable, we have the following:

[1] One might also settle for committees containing limited number edges; independent set takes this view to the extreme.

Corollary 1. CONN-CC (CF-CC) *is* NP-complete *and* W[2]-hard *wrt k and* CONN-GS (CF-GS) *is* NP-complete *and* W[1]-hard *wrt k, even when the candidate graph is a clique (respectively, edgeless).*

1.2 Paper Structure

Throughout, we study the CF-$\mathcal{R}$ and CONN-$\mathcal{R}$ problems for $\mathcal{R} \in \{\text{CC}, k\text{-Borda}, \text{GS}\}$. As CC, GS, and finding an independent set of size k are all computationally intractable, we indeed expect most of our problems to be intractable. In fact even though both CONNECTIVITY and k-BORDA are polynomial-time solvable, separately, we nevertheless show that CONN-k-BORDA is NP-complete. We thus concentrate on various ways of breaking this computational intractability: In particular, we use parameterized complexity, domain restrictions (restricting voter preferences), and graph classes (restricting allowed candidate graphs).

We concentrate on natural election parameters, frequently studied in papers on parameterized complexity of computational social choice problems: The number of candidates, m, is frequently small – e.g., in most political elections; the number of voters, n, is sometimes small – e.g., when selecting a jury, judges for a talent competition, or board of directors (see, e.g., [Chen *et al.*, 2015]); and the committee size, k, is small in many real-life scenarios such as in shortlisting tasks, parliamentary elections, and forming a rescue team.

We consider single-peaked profiles, unanimous profiles, and strongly unanimous profiles as domain restrictions; while single-peaked is probably the most popular domain restriction, the other two domain restrictions, though perhaps too restricted to be practically useful, nevertheless provide good insights on the combinatorics of our problems. Additionally, we study our problems on inputs where the candidate graph has a certain structure.

The paper is structured by problems: In Sections 2.1, 2.2, and 2.3 we study connected committees for CC, k-Borda, and GS, respectively; while in Sections 3.1, 3.2, and 3.3 we study conflict-free committees for CC, k-Borda, and GS, respectively. To give a full picture, Table 1 lists all our results. We highlight our most technically-involved results: **Theorems 1, 4, and 13**. As the table shows, we are able to identify several cases for which efficient algorithms exist for our problems. We mention that, while doing so, we also prove certain results regarding combinatorial problems on graphs that might be of independent interest. Some proof details are missing due to limited space. The notation $\mathcal{O}^\star(f(k))$ suppresses $\text{poly}(n)$ factors.

2 Connected Committees

We study the complexity of finding connected committees.

2.1 Connected-CC

Here, we first give an FPT algorithm wrt $n+k$. Towards this, we first observe that in polynomial time, MAX-CONN-CC can be reduced to BUDG-MAXWT-TST[2], in which given an edge-weighted graph G, a set of terminals T, and an integer t; we look for a maximum weight tree of size t containing each of the terminals as leaves, where the size of the tree in the reduced instance, that is t, is $n+k$. We design an FPT algorithm wrt t for BUDG-MAXWT-TST which can then be used to solve MAX-CONN-CC.

Our FPT algorithm for BUDG-MAXWT-TST with tree size as the parameter relies on the following result; unlabeled graph is one in which the vertices are not named.

Proposition 1 ([Beyer and Hedetniemi, 1980; Otter, 1948])**.** *There are $t_n = 2.956^n$ non-isomorphic unlabeled trees on n vertices. Moreover, we can enumerate them in time $\mathcal{O}(t_n)$.*

Theorem 1. BUDG-MAXWT-TST *is* FPT *wrt t. Consequently,* MAX-CONN-CC *is* FPT *wrt $n+k$.*

Proof. Let (G, T, t) be the given instance of BUDG-MAXWT-TST. Let $T^\star$ be a solution to BUDG-MAXWT-TST. Using Proposition 1 we enumerate all unlabeled trees T_{ul} on t nodes. For each of these trees, if the number of its leaves is less than $|T|$, then we discard the tree and move to the next one. For each unlabeled tree T_{ul}, we recast it as the MAX-EDGEWT-TREE-ISO problem as follows: Given an edge-weighted graph G and a tree T_{ul}, we find a maximum edge-weighted tree in G that is isomorphic to T_{ul}. For BUDG-MAXWT-TST, we go through each of the unlabeled trees as a potential T_{ul} and apply the algorithm for MAX-EDGEWT-TREE-ISO with G and T_{ul} as the input. This step takes $2.956^t t^{\mathcal{O}(1)}$ time. By our choice we know that T_{ul} has at least $|T|$ leaves. We first guess an injective mapping from the terminals to the leaves of T_{ul} (there are at most $t^{|T|} = 2^{\mathcal{O}(t \log t)}$ guesses, in total). Let this mapping be g. Now our problem reduces to MAX-EDGEWT-TREE-ISO where some leaves are already labeled. This can be solved using the algorithm for k-TREE [Fomin *et al.*, 2016] in $2.6182^{t-|T|} t^{\mathcal{O}(1)}$ time. (Note that the algorithm given in [Fomin *et al.*, 2016] is for finding a minimum weight isomorphic copy. However, replacing minimum weight q-representative with maximum weight q-representative in that algorithm will yield the maximization version of the problem.) This implies that the total running time of our algorithm is upper bounded by the number of unlabeled trees on t vertices, the number of injective mapping from terminals to the leaves of T_{ul} and the algorithm for MAX-EDGEWT-TREE-ISO, where the leaves are already labeled. Thus, the total time required is $2^{\mathcal{O}(t \log t)} t^{\mathcal{O}(1)}$. Thus, we can conclude that MAX-CONN-CC is FPT wrt $n+k$. □

While CC is solvable in polynomial time for single-peaked (SP) profiles[3] [Betzler *et al.*, 2013], CONN-CC is not.

Theorem 2. CONN-CC *is* NP-complete *for SP profiles.*

This leads us to a more restricted domain, namely *unanimous* profiles, in which every voter rank the same candidate on top, and *strongly unanimous* (SU) profiles, in which every voter has the same preference list. We have the following:

[2] It is NP-hard (since TERMINAL STEINER TREE, in which we look for a minimum cost terminal Steiner tree, is NP-hard [Lin and Xue, 2002]).

[3] It is a domain restriction, in which every voter is single-peaked wrt an axis π, i.e, for every voter v, and for each pair of candidates a, b, if either $a \; \pi \; b \; \pi \; top(v)$ or $top(v) \; \pi \; b \; \pi \; a$ holds, where $top(v)$ is the first ranked candidate of voter v, then v ranks b above a.

	Connected			**Conflict-free**		
	Complexity	Restricted Candidate Graph/Election	Ref.	Complexity	Restricted Candidate Graph/Election	Ref.
CC	NPC and W-h wrt k	H is clique	Cor. 1	NPC and W-h wrt k	H is edgeless	Cor. 1
	FPT wrt $n+k$	🟪	Thm. 1	NPC and W-h wrt k	$n=1, s=0$	Thm. 6
	NPC	SP	Thm. 2	FPT wrt. n	$\lvert\mathcal{F}\rvert$ in H is polyn.	Thm. 7
	NPC and W-h wrt k	unanimous	Obs. 2	$\mathcal{O}^\star(1.1996^m)$ solvable	constant n	Thm. 8
	P	strongly unanimous	Obs. 2	$\mathcal{O}^\star(1.4423^m)$ solvable	SP	Thm. 9
	FPT wrt $\ell+k, q+k$	🟪	Lem. 1			
	Open wrt n					
	Open wrt k	**SP**				
k**-Borda**	NPC	🟪	Thm. 3	NPC and W-h	$n=1, s=0$	Thm. 10
	FPT wrt k		Thm. 4	$\mathcal{O}^\star(1.4423^m)$ solvable	🟪	Thm. 11
	Open wrt n					
GS	NPC and W-h wrt k	H is clique	Cor. 1	NPC and W-h wrt k	H is edgeless	Cor. 1
	$\mathcal{O}^\star(2^h)$ solvable	🟪	Thm. 5	NPC and W-h wrt k	$n=2$, SP	Thm. 12
	NPC and W-h wrt k	unanimous	Obs. 3	$\mathcal{O}^\star(1.4656^m)$ solvable	🟪	Thm. 13
	P	strongly unanimous	Obs. 3	$\mathcal{O}^\star(1.46^h)$ solvable		Thm. 14
				NPC and W-h wrt k	unanimous	Obs. 4
				P	strongly unanimous	Obs. 4

Table 1: Summary of our results. H denotes the candidate graph. Parameter k is the size of the committee, n is #voters, m is #candidates, s is the committee score, ℓ (or q) is the minimum number of voters that need to change their preference list (respectively, minimum number of consecutive swaps required) so that the profile is strongly unanimous, and h is the size of the tournament vertex deletion set. $\mathcal{F}$ is the set of all maximal independent sets in H. A **purple cell** implies that the result holds for an arbitrary graph and election. SP stands for single-peaked. NPC and W-h stands for NP-completeness and W-hardness, respectively.

Observation 2. CONN-CC *is* NP-complete *and* W[2]-hard *wrt k even for unanimous profiles. But, for SU profiles,* MAX-CONN-CC *can be solved in polynomial time.*

As SU profiles are very restricted, we view them as trivial instances and study two related distance parameters.

Lemma 1. MAX-CONN-CC *is* FPT *wrt $\ell + k$ and $q + k$, where ℓ is the minimum number of voters that need to change their preference list, and q is the minimum number of consecutive swaps required, so that the profile is SU.*

2.2 Connected-k-Borda

Even though k-Borda is polynomial-time solvable, we have the following for CONN-k-BORDA:

Theorem 3. CONN-k-BORDA *is* NP-complete.

To break the intractability, we first consider parameter k. Towards this, we first observe that in polynomial time, MAX-CONN-k-BORDA can be reduced to MAX-NODEWT-k-TREE, in which we are given a node-weighted graph G and need to find a k-sized tree of maximum weight. Given an instance (E, H, k, s) of CONN-k-BORDA, we create an instance $(G = H, k' = k, w = s)$ of MAX-NODEWT-k-TREE. The weight of each vertex is set to be the Borda score of the corresponding candidate.

Theorem 4. MAX-NODEWT-k-TREE *is* FPT *wrt k. Consequently,* MAX-CONN-k-BORDA *is* FPT *wrt k.*

Proof. Our proof combines three ingredients: (i) enumeration of unlabeled trees, (ii) the color-coding technique of Alon et al. [1995], and (iii) dynamic programming. Let T denote a solution to MAX-NODEWT-k-TREE. Observe that if we ignore the labels of the vertices of T, then it is isomorphic to an unlabeled tree, say T_{ul} on k vertices and hence T_{ul} will be enumerated when we enumerate all unlabeled trees on k vertices, using Proposition 1. Thus, in $\mathcal{O}(2.956^k k^{\mathcal{O}(1)})$ time, we have reduced MAX-NODEWT-k-TREE to a problem of finding a tree in G that is isomorphic to a given tree and has maximum weight – we refer to this problem as MAX-NODEWT-TREE-ISO. That is, given a vertex-weighted graph G and a tree T_{ul}, find a maximum weight tree in G that is isomorphic to T_{ul}.

Algorithm for MAX-NODEWT-TREE-ISO[4]: Rather than using the randomized version of color-coding [Alon *et al.*, 1995] for our problem, we directly apply the deterministic version. To that end we use the (n, k)*-perfect hash functions*.

Definition 2. *A family of functions $f_1, \ldots, f_t$ from a universe U of size n to a universe of size k is a (n, k)-perfect family of hash functions if for every set $S \subseteq U$ such that $|S| = k$ there exists an i such that the restriction of f_i to S is injective.*

We need the following result:

Proposition 2 ([Naor *et al.*, 1995])**.** *For any $n, k \geq 1$ one can construct an (n, k)-perfect hash family of size $e^k k^{\mathcal{O}(\log k)} \log n$ in time $e^k k^{\mathcal{O}(\log k)} n \log n$.*

Recall that T_{ul} is an input to MAX-NODEWT-TREE-ISO and T is a hypothetical solution to the problem. To give an algorithm for the MAX-NODEWT-TREE-ISO problem, we convert this into a colored problem by applying Proposition 2.

[4] Due to space restriction, we give here slightly sub-optimal algorithm that is easier to explain and sufficient for the result, however, one can design a $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ time algorithm using the notion of representative families.

Note that by the definition of (n,k)-perfect family of hash functions, there exists a function, say f, such that the restriction of f to the vertices of $V(T)$ is injective. By trying all possible functions in the hash family, we find the desired one. Let us fix such a function, f. This partitions the vertex set $V(G)$ into $V_i = f^{-1}(i)$, $i \in [k]$. We refer to each V_i as a *color class*.

To find a subgraph T of G of maximum weight that is isomorphic to T_{ul}, we "guess", for every vertex of T_{ul}, the color class that contains this vertex. This corresponds to $k! = 2^{\mathcal{O}(k \log k)}$ guesses. Let this injective (coloring) function be $g : V(T_{\mathsf{ul}}) \rightarrow [k]$. Given the function g, we root the tree T_{ul} at some arbitrary vertex, say r, and assign to each vertex the name of the color class. Denote this tree by $T_{\mathsf{ul}}^{\mathsf{col}}$. We will use $T_{\mathsf{ul}}^{\mathsf{col}}$ to do a bottom-up dynamic programming from leaves to root. We first perform the following cleaning operation in the graph G: Delete any edge from G that is not a potential edge of T; i.e., delete an edge from G with colored end-points as $i, j \in [k]$, if the edge $ij \notin T_{\mathsf{ul}}^{\mathsf{col}}$. We delete all the vertices of degree 0 from G. Next, we go bottom up and delete vertices from color classes that cannot be mapped to T. For a color class V_i such that i is a leaf in $T_{\mathsf{ul}}^{\mathsf{col}}$, we do nothing. Now for every vertex j in $T_{\mathsf{ul}}^{\mathsf{col}}$, let $j_1, j_2, \ldots, j_\ell$ be the children of j. We keep a vertex $v \in V_j$ only if it has at least one neighbor in V_{j_s}, for all $s \in [\ell]$.

Observe that there is a tree in G that is isomorphic to T_{ul} iff each color class is non-empty. To find the maximum weight tree isomorphic to T_{ul}, rather $T_{\mathsf{ul}}^{\mathsf{col}}$, we do dynamic programming as follows: For a vertex $q \in V(T_{\mathsf{ul}}^{\mathsf{col}})$, let $T_{\mathsf{ul,q}}^{\mathsf{col}}$ denote the subtree of $T_{\mathsf{ul}}^{\mathsf{col}}$ rooted at q. For a vertex $v \in V_q$, $A[v,q]$ stores the maximum weight of a subtree in G that is rooted at v and isomorphic to $T_{\mathsf{ul,q}}^{\mathsf{col}}$. Clearly, if q is a leaf of $T_{\mathsf{ul}}^{\mathsf{col}}$, then $A[v,q] = w(v)$. Else, we can compute this using the given recurrence: Let q_i, for each $i \in [\ell]$ be the children of q, then: $A[v,q] = w(v) + \sum_{j=1}^{\ell} \max \left\{ A[u, q_j] : u \in N(v) \cap V_{q_j} \right\}$.

The correctness of the recursive relations follows from the disjointness provided by the function g and the cleaning process. The running time of the algorithm is upper bounded by the product of tree enumeration time, number of hash-functions f, number of functions g, and the dynamic programming; $\mathcal{O}^\star(2^{\mathcal{O}(k \log k)})$ in total. □

2.3 Connected-Gehrlein Stable

Here, we obtain some tractability results. To define our parameter, we need the known graph theoretic formulation of GS. Given an election E, the *majority graph* consists of vertices corresponding to every candidate in the election and if more than half of the voters prefer candidate u over candidate v in the pairwise election between u and v, then there is an arc from u to v. Note that the task of finding a GS committee for (E,k) boils down to finding a k-sized subset S in the majority graph such that no vertex outside S has an out-neighbor in S. When the majority graph is a tournament (i.e., a directed graph in which there is an arc between every pair of vertices), GS has a unique solution, which can be found in polynomial time [Aziz *et al.*, 2017b]. Thus, CONN-GS can be solved in polynomial time as we only need to check connectivity in the solution of GS. As CONN-GS is tractable on tournaments, we study CONN-GS wrt the number h of vertices whose deletion from the majority graph leads to a tournament; we refer to h as the size of *tournament vertex deletion set* (tvd). Note that h is smaller than the number of pairs of candidates that are in tie. In real-life scenarios, it is usually unlikely to have ties between lots of candidates. The algorithm for the following result is similar to the algorithm in [Gupta *et al.*, 2019], who study h as a parameter for GS.

Theorem 5. CONN-GS *can be solved in* $\mathcal{O}^\star(2^h)$ *time, where* h *is the size of the minimum tournament vertex deletion set.*

We have the following result for a restricted domain.

Observation 3. CONN-GS *is* NP-complete *and* W[1]-hard *wrt* k *even for unanimous profile. But, the problem is polynomial-time solvable for SU profiles.*

3 Conflict Free Committees

We study the complexity of finding conflict-free committees. Most of the NP-hard and W[1]-hard reductions are from INDEPENDENT SET (IS, in short) which is known to be NP-hard [Garey and Johnson, 2002] and W[1]-hard [Downey and Fellows, 1995].

3.1 Conflict Free-Chamberlin Courant

Here, even the parameterization by the number n of voters and the committee score s does not break the intractability, as can be exhibited by a reduction from IS.

Theorem 6. CF-CC *is* NP-complete *and* W[1]-hard *wrt* k *even when there is one voter and the score is* 0.

Next, we describe an FPT algorithm wrt n for special classes of candidate graphs, in particular, when the number of maximal independent sets in the candidate graph is polynomial (There are graph classes with bounded number of maximal independent sets; most notable is the class of split graphs, in which the number of maximal independent sets is linear in the number of vertices). The algorithm is based on the idea that any solution of CF-CC is a subset of a maximal independent set in the candidate graph. Therefore, we can restrict the election on the candidates to a maximal independent set of H and find a CC committee in this restricted election. Thus, if the number of maximal independent sets is polynomial in the input size, then it can be enumerated in polynomial time [Johnson *et al.*, 1988], and hence we can use the known FPT algorithm for CC wrt n. Thus, we have the following:

Theorem 7. *If the number of maximal independent sets in the candidate graph* H *is polynomial, then* MAX-CF-CC *is* FPT *wrt* n*.*

Turns out that we can improve the brute-force algorithm for elections with a constant number of voters.

Theorem 8. MAX-CF-CC *can be solved in* $\mathcal{O}^\star(1.1996^m)$ *time for a constant number of voters.*

Using the idea of enumerating maximal independent sets in the candidate graph, we have the following:

Theorem 9. MAX-CF-CC *can be solved in* $\mathcal{O}^\star(1.4423^m)$ *time for SP profiles.*

3.2 Conflict Free-k-Borda

We first have the following intractability result which can be exhibited by a reduction from IS.

Theorem 10. CF-k-BORDA *is* NP-complete *and* W[1]-hard *wrt k even when there is one voter and score is* 0.

Using the same intuition as for Theorem 7 that the solution is contained in a maximal independent set of H, and the polynomial time algorithm for k-BORDA, we have the following:

Theorem 11. MAX-CF-k-BORDA *can be solved in* $\mathcal{O}^\star(1.4423^m)$ *time.*

3.3 Conflict Free-Gehrlein Stable

Here, we obtain that CF-GS is NP-hard even for two voters which can be exhibited by a reduction from IS.

Theorem 12. CF-GS *is* NP-complete *and* W[1]-hard *wrt k even when there are two voters and the profile is SP.*

Next we describe an FPT algorithm wrt the number m of candidates that is better than the brute-force algorithm. To this end, we consider the graph-theoretic formulation of GS (see Section 2.3). Let (M, H, k) be an instance of CF-GS, where M is the majority graph, H is the candidate graph, and k is the size of the committee. The algorithm is based on the following observations that ensure that committee is GS.

1. If a candidate u is in the committee, then its *in-reachability set* is also in the committee (a vertex v is in the in-reachability set of u, if there is a directed path from v to u).
2. If a candidate u is not in the committee, then its *out-reachability set* is also not in the committee (a vertex v is in the out-reachability set of u, if there is a directed path from u to v).
3. If u and v are neighbors in H, and u is in the in-reachability set of v, then v cannot be in the solution.
4. If u is in the committee, then none of its neighbors in H is in the committee.

Following (2) and (3), we apply the following reduction rule. We reuse the notation after application of every reduction rule (RR) and branching rule (BR), and denote the reduced instance by (M, H, k).

RR 1. *Let $uv \in E(H)$. Suppose that u is in the in-reachability set of v in M. Then, delete v and its out-reachability set from M and H.*

We define the in-reachability set of a subset of vertices X, denoted by $R^-(X)$, as the set containing the in-reachable set of every vertex $x \in X$. Similarly, define the out-reachability set of a subset of vertices X, and denote it by $R^+(X)$. Next, we apply several branching rules.

BR 1. *Suppose that u is vertex of degree at least 2 in H. Then, branch on u as follows:*

1. u **is in the solution.** *We add $\{u\} \cup R^-(u)$ to the solution and delete $\{u\} \cup R^-(u)$, $N(\{u\} \cup R^-(u))$, and $R^+(N(\{u\} \cup R^-(u)))$ (out-reachability set of $N(\{u\} \cup R^-(u))$) from M and H and obtain the majority graph M' and the candidate graph H'. Recurse on the instance $(M', H', k - |\{u\} \cup R^-(u)|)$.*

2. u **is not in the solution.** *We delete $\{u\}$ and $R^+(u)$ from M and H and obtain the majority graph M' and the candidate graph H'. Recurse on the instance (M', H', k).*

BR 2. *Suppose that u is a vertex with at least one in-neighbor and at least one out-neighbor in M. Then, branch on u in a similar way as in BR 1.*

BR 3. *Suppose that u is a vertex with either at least two in-neighbors or at least two out-neighbors in M. Then, branch on u in a similar way as in BR 1.*

After applying BR 1 to 3, exhaustively, M consists of isolated vertices and disjoint arcs (i.e., arcs that do not share their end-points), and in H, each vertex has degree at most one. Therefore, if M contains at least $2k$ isolated vertices or if the end-points of the disjoint arcs in M are isolated vertices in H, then we can solve the instance in polynomial time. Otherwise, we branch as follows.

BR 4. *Suppose that (u, v) is an arc in M. If either u or v has degree 1 in H, then branch on v in a similar way as in BR 1.*

The running time of the whole algorithm is governed by the recurrence $T(m) \leq T(m-1) + T(m-3)$, which solves to $\mathcal{O}^\star(1.4656^m)$. Thus, we have the following:

Theorem 13. CF-GS *can be solved in* $\mathcal{O}^\star(1.4656^m)$ *time.*

Next, we study CF-GS wrt tvd and have the following:

Theorem 14. CF-GS *can be solved in* $\mathcal{O}^\star(1.46^h)$ *time, where h is the size of tournament vertex deletion set in M.*

We have the following result in restricted domain:

Observation 4. CF-GS *is* NP-complete *and* W[1]-hard *wrt k even for unanimous profiles. But, the problem is polynomial-time solvable when the profile is SU.*

4 Future Direction

Beside considering other ways of breaking the computational intractability (e.g., using randomization, approximation algorithms, and heuristic methods) and studying further multiwinner rules $\mathcal{R}$ and graph properties $\mathcal{Q}$, further research directions include studying other ways of combining multiwinner rules $\mathcal{R}$ and graph properties $\mathcal{Q}$, such as by aiming at finding points on the Pareto curve (similarly in spirit to multigoal committee selection [Kocot *et al.*, 2019]) or, e.g., relaxing the requirement of Independent Set by merely minimizing the number of edges between committee members.

Acknowledgments

We are grateful to Dr. Nimrod Talmon for fruitful discussions. Sushmita Gupta was supported by SERB-Starting Research Grant (SRG/2019/001870). Saket Saurabh was supported by European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant no. 819416), and Swarnajayanti Fellowship grant (DST/SJF/MSA- 01/2017-18).

References

[Alon *et al.*, 1995] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.

[Aziz *et al.*, 2017a] Haris Aziz, Marcus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. Justified representation in approval-based committee voting. *Soc. Choice Welf.*, 48(2):461–485, 2017.

[Aziz *et al.*, 2017b] Haris Aziz, Edith Elkind, Piotr Faliszewski, Martin Lackner, and Piotr Skowron. The condorcet principle for multiwinner elections: From shortlisting to proportionality. *arXiv preprint arXiv:1701.08023*, 2017.

[Aziz *et al.*, 2017c] Haris Aziz, Edith Elkind, Piotr Faliszewski, Martin Lackner, and Piotr Skowron. The condorcet principle for multiwinner elections: From shortlisting to proportionality. In *Proceedings of IJCAI 2017*, pages 84–90, 2017.

[Betzler *et al.*, 2013] Nadja Betzler, Arkadii Slinko, and Johannes Uhlmann. On the computation of fully proportional representation. *J. Artif. Intell. Res.*, 47:475–519, 2013.

[Beyer and Hedetniemi, 1980] Terry Beyer and Sandra Mitchell Hedetniemi. Constant time generation of rooted trees. *SIAM J. Comput.*, 9(4):706–712, 1980.

[Brandt *et al.*, 2016] Felix Brandt, Vincen. Conitzer, Ulle Endriss, Jerôme Lang, and Ariel. D. Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[Bredereck *et al.*, 2018] R. Bredereck, Piotr Faliszewski, Ayumi Igarashi, Martin Lackner, and Piotr Skowron. Multiwinner elections with diversity constraints. In *Proceedings of AAAI '18*, pages 933–938, 2018.

[Brill *et al.*, 2019] Marcus Brill, Piotr Faliszewski, Frank Sommer, and Nimrod Talmon. Approximation algorithms for balancedcc multiwinner rules. In *Proceedings of AAMAS '19*, pages 494–502, 2019.

[Chen *et al.*, 2015] Jiehua Chen, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Elections with few voters: Candidate control can be easy. In *Proceedings of AAAI '15*, pages 2045–2051, 2015.

[Downey and Fellows, 1995] Robert G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness II: On completeness for W [1]. *Theor. Comput. Sci.*, 141(1-2):109–131, 1995.

[Elkind *et al.*, 2017] Edith Elkind, Piotr Faliszewski, Piotr Skowron, and Arkadii Slinko. Properties of multiwinner voting rules. *Soc. Choice Welf.*, 48(3):599–632, 2017.

[Endriss, 2017] Ulle Endriss. *Trends in Computational Social Choice*. AI Access Foundation, 2017.

[Faliszewski *et al.*, 2016] Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. Committee scoring rules: Axiomatic classification and hierarchy. In *Proceedings of IJCAI '16*, pages 250–256, 2016.

[Faliszewski *et al.*, 2017a] Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. Multiwinner voting: A new challenge for social choice theory. In U. Endriss, editor, *Trends in Computational Social Choice*. AI Access Foundation, 2017.

[Faliszewski *et al.*, 2017b] Piotr Faliszewski, Piotr Skowron, and Nimrod Talmon. Bribery as a measure of candidate success: Complexity results for approval-based multiwinner rules. In *Proceedings of AAMAS'17*, pages 6–14, 2017.

[Faliszewski *et al.*, 2018] Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. Multiwinner analogues of the plurality rule: axiomatic and algorithmic perspectives. *Soc. Choice Welf.*, 51(3):513–550, 2018.

[Fomin *et al.*, 2016] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016.

[Garey and Johnson, 2002] Michael R. Garey and David S. Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.

[Gupta *et al.*, 2019] Sushmita Gupta, Pallavi Jain, Sanjukta Roy, Saket Saurabh, and Meirav Zehavi. Gehrlein stability in committee selection: Parameterized hardness and algorithms. In *Proceedings of AAMAS'19*, pages 511–519, 2019.

[Izsak *et al.*, 2018] Rani Izsak, Nimrod Talmon, and Gerhard J. Woeginger. Committee selection with intraclass and interclass synergies. In *Proceedings of AAAI '18*, pages 1071–1078, 2018.

[Johnson *et al.*, 1988] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. On generating all maximal independent sets. *Inf. Process. Lett.*, 27(3):119–123, 1988.

[Kocot *et al.*, 2019] Maciej Kocot, Anna Kolonko, Edith Elkind, Piotr Faliszewski, and Nimrod Talmon. Multigoal committee selection. In *Proceedings of IJCAI '19*, pages 250–256, 2019.

[Lin and Xue, 2002] Guohui Lin and Guoliang Xue. On the terminal steiner tree problem. *Inf. Process. Lett.*, 84(2):103–107, 2002.

[Naor *et al.*, 1995] Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *Proceedings of FOCS'95*, pages 182–191, 1995.

[Otter, 1948] Richard Otter. The number of trees. *Ann. Math.*, pages 583–599, 1948.

[Procaccia *et al.*, 2007] Ariel D. Procaccia, Jeffrey S. Rosenschein, and Aviv Zohar. Multi-winner elections: Complexity of manipulation, control and winner-determination. In *Proceedings of IJCAI '07*, volume 7, pages 1476–1481, 2007.

[Procaccia *et al.*, 2008] Ariel D. Procaccia, Jeffrey S. Rosenschein, and Aviv Zohar. On the complexity of achieving proportional representation. *Soc. Choice Welf.*, 30(3):353–362, 2008.

[Talmon, 2018] Nimrod Talmon. Structured proportional representation. *Theor. Comput. Sci.*, 708:58–74, 2018.