Gehrlein Stability in Committee Selection: Parameterized Hardness and Algorithms

Sushmita Gupta
National Institute for Science Education and Research, Bhubaneswar, India
sushmitagupta@niser.ac.in

Pallavi Jain
The Institute of Mathematical Sciences, HBNI, Chennai, India
pallavij@imsc.res.in

Sanjukta Roy
The Institute of Mathematical Sciences, HBNI, Chennai, India
sanjukta@imsc.res.in

Saket Saurabh
The Institute of Mathematical Sciences, HBNI, Chennai, India
University of Bergen, Bergen, Norway
saket@imsc.res.in

Meirav Zehavi
Ben-Gurion University, Beersheba, Israel
meiravze@bgu.ac.il

ABSTRACT

In a multiwinner election based on the Condorcet criterion, we are given a set of candidates, and a set of voters with strict preference ranking over the candidates. A committee is *weakly Gehrlein stable* (WGS) if each committee member is preferred to each non-member by at least half of the voters. Recently, Aziz et al. [IJCAI 2017] studied the computational complexity of finding a WGS committee of size k. They show that this problem is NP-hard in general and polynomial time solvable when the number of voters is odd. In this article, we initiate a systematic study of the problem in the realm of parameterized complexity. We first show that the problem is W[1]-hard when parameterized by the size of the committee. To overcome this intractability result, we use a known reformulation of WGS as a problem on directed graphs and then use parameters that measure the "structure" of these directed graphs.

In particular, we consider the majority graph, defined as follows: there is a vertex corresponding to each candidate, and there is a directed arc from a candidate c to c' if the number of voters that *prefer* c over c' is more than those that prefer c' over c. The problem of finding WGS committee of size k corresponds to finding a vertex subset X of size k in the majority graph with the following property: the set X contains no vertex outside the committee that has an in-neighbor in X. Observe that the polynomial time algorithm of Aziz et al. [IJCAI 2017] corresponds to solving the problem on a tournament (a complete graph with orientation on edges). Thus, natural parameters to study our problem are "closeness" to being a tournament. We define closeness as the number of missing arcs in the given directed graph and the number of vertices we need to delete from the given directed graph such that the resulting graph is a tournament. We show that the problem is fixed parameter tractable (FPT) and admits linear kernels with respect to closeness parameters. Finally, we also design an exact exponential time algorithm running in time $O(1.2207^n n^{O(1)})$. Here, n denotes the number of candidates.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

KEYWORDS

committee selection; social choice; parameterized complexity

ACM Reference Format:
Sushmita Gupta, Pallavi Jain, Sanjukta Roy, Saket Saurabh, and Meirav Zehavi. 2019. Gehrlein Stability in Committee Selection: Parameterized Hardness and Algorithms. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

1 INTRODUCTION

An important question in social choice theory is—"how to choose a non-controversial committee of size k?" Such a question arises while electing parliaments in modern democracies, selecting a group of representatives in an organisation, in taking business decisions or shortlisting tasks. In voting theory, all these scenarios can be captured by *multiwinner elections*. In particular, the problem of selecting a committee can be formulated as follows. Given a set of candidates and a set of voters with strict preference ranking over the candidates, find a committee of size k satisfying certain acceptability criteria. However, what acceptability criteria should be chosen? For a single winner election, Condorcet [7] suggested that some candidate can be considered a winner if s/he is preferred by at least half of the voters over every other candidate. Of course, such a candidate may not exist. Fishburn [17] generalized the idea of Condorcet for a single winner election to a committee. This definition of *Condorcet committee* requires that each voter has explicit preferences over the committees, or there is some way to infer these preferences. According to Fishburn, a committee is a Condorcet committee if it is preferred by at least half of the voters over any other committee of the same size. Darmann [9] defined two notions of Condorcet committee, *weak* and *strong*, where preferences over the committees are implicit. Specifically, a committee of a given size k is weak (strong), if it is at least as good as (better than) any other committee of size k in a pairwise majority comparison. The problems corresponding to finding a weak (strong) Condorcet committee of size k are Weak (Strong) Condorcet k-Committee.

Gerhlein [19] defined a new notion of Condorcet committee by considering each candidate of the committee instead of whole committee. According to his definition, a committee is Condorcet if a candidate in the committee is preferred by at least half of the voters to each non-member. Note that such a committee might not exist. Moreover, a committee is called *weakly* (*strongly*) *Gehrlein-stable* if every candidate c in the committee is preferred by at least (more than) half of the voters in the pairwise election between c and every non-committee member d. We would like to point out here that when the number of voters is odd, weakly and strongly Gehrlein-stable committee are equivalent. However, this is not the case when the number of voters is even. We remark that, in the literature, there are also other ways of comparing a committee with other committees [4, 10, 12, 13, 23].

For the committee selection problem, extensive research has been conducted to study voting rules and their stability in the context of selecting a committee [6, 11, 22, 26]. Darmann [9] analyzed the computational complexity of Weak and Strong Condorcet k-Committee. He studied the problem with different voting rules, including Borda voting, plurality voting, antiplurality voting, and t-approval, where $t \geq 2$. Specifically, he proved that Weak and Strong Condorcet k-Committee are coNP-hard under Borda and 2-approval voting schemes. Furthermore, he showed that Weak and Strong Condorcet k-Committee are polynomial time solvable under plurality and antiplurality voting schemes. For more literature on multiwinner elections, we refer to [15].

Recently, Aziz et al. [1, 2] studied the computational complexity of finding a Gehrlein-stable committee of size k. They proved that finding a strongly Gehrlein-stable committee of size k (and determining that one exists) can be done in polynomial time. However, computing a weakly Gehrlein-stable committee of size k is NP-hard. Aziz et al. [1, 2] proposed to study this problem from the perspective of parameterized complexity and exact exponential time algorithms. In this article, we initiate a systematic study of finding a weakly Gehrlein-stable committee of size k in the realm of parameterized complexity. We call this problem as Gehrlein Stable Committee Selection (GSCS).

We first show that GSCS is W[1]-hard when parameterized by the size of the committee. That is, the problem is unlikely to admit an algorithm with running time $f(k)n^{O(1)}$. To overcome this intractability result, we seek relevant alternate parameters that could lead to tractable algorithms. To achieve this, we consider a model of GSCS as a problem on directed graphs and then use parameters that measure the "structure" of these directed graphs. In particular, we consider the majority graph [25], defined as follows. Given any election $\mathcal{E} = (C, \mathcal{V})$, we define the majority graph $\mathcal{M}_{\mathcal{E}} = (C, \mathcal{A})$ on the vertex set C and an arc $(c, c') \in \mathcal{A}$ if and only if candidate c is more popular than c' in the election $\mathcal{E}$ (denoted by $c >_{\mathcal{E}} c'$). In other words, $(c, c') \in \mathcal{A}$ if and only if the number of voters that *prefer* c over c' is strictly more than those preferring c' over c. For example, Figure 1 illustrates the majority graph corresponding to the election in Table 1.

Now, note that if a committee S of size k is stable in an election $\mathcal{E} = (C, \mathcal{V})$, then there does not exist a candidate $u \in C \setminus S$ that is preferred over any candidate in S in the pairwise election between y and the candidates of S. This implies that there does not exist a vertex $u \in C \setminus S$ such that $(u, v) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$ for some $v \in S$. Hence, for any $v \in S$, all the in-neighbors of v in the graph $\mathcal{M}_{\mathcal{E}}$ belong to S. Thus, the problem of finding WGS committee of size k corresponds to finding a vertex subset X of size k in the majority graph with the following property: the set X contains no vertex outside the committee that has an in-neighbor in X. We will use this formulation of the problem in the paper. In particular, we will study the following problem.

> Gehrlein Stable Committee Selection (GSCS)
> **Input:** A majority graph $\mathcal{M}_{\mathcal{E}} = (C, \mathcal{A})$ for an election $\mathcal{E}$ and a positive integer k such that $k \leq |C|$.
> **Question:** Does there exist a subset of vertices $S \subseteq C$, $|S| = k$ such that for every $v \in S$, each of the in-neighbors of v (if any) lies in S?

Such a set S is a *solution* to the problem.

Our algorithmic contributions. One way to discover relevant parameters for studying a graph problem is to find a family of graphs, say $\mathcal{F}$, where the problem is polynomial time solvable; then, the problem is studied with an *edit distance*—the number of vertices/edges deleted (or edges added) to transform the input graph into a graph in $\mathcal{F}$—as a parameter. Aziz et al. [1, 2] showed that GSCS is polynomial time solvable when the number of voters is odd. Observe that the polynomial time algorithm of Aziz et al. [1, 2] corresponds to solving the problem on a tournament. Thus, natural parameters to study our problem correspond to vertex/edge editing operations into the family of tournaments. We study GSCS with two "editing parameters": the number of missing arcs in the given directed graph (l) and the size of a *tournament vertex deletion set (***tvd***)* (q)—that is, a subset of vertices whose deletion from the given directed graph results in a tournament. The number l corresponds to the number of pairs of candidates which are *tied* among each other in the pairwise election and q could be thought of as the smallest subset of candidates who are in a tie with some candidate(s) such that the deletion of this subset will render the resulting majority graph a tournament. Since **tvd** is smaller than the number of candidates who are in a tie, it makes **tvd** a natural parameter to study from a computational perspective. We show that the problem is fixed parameter tractable (FPT) and admits linear kernels with respect to the parameters l and q. In particular, we obtain the following results.

- GSCS can be solved optimally in $O^{\star}(1.2207^n)$[1] time. Here, n denotes the number of candidates ($|C|$). This resolves a question asked in the conclusion of Aziz et al. [2].
- GSCS admits an FPT algorithm with running time $O^{\star}(1.2738^q)$.
- GSCS admits a kernel with $4q + 1$ vertices. That is, there a is a polynomial time algorithm that given an instance $(\mathcal{M}_{\mathcal{E}}, k)$ of GSCS returns an instance $(\mathcal{M}'_{\mathcal{E}}, k')$ of GSCS such that $(\mathcal{M}_{\mathcal{E}}, k)$ is a yes-instance if and only if $(\mathcal{M}'_{\mathcal{E}}, k')$ is a yes-instance and $|V(\mathcal{M}'_{\mathcal{E}})| \leq 4q + 1$.
- GSCS admits an algorithm with running time $O^{\star}(1.2207^l)$ and has a kernel with $2l + 1$ vertices. These results are obtained as corollaries to the results for the parameter q.

2 PRELIMINARIES

The set $\{1, \ldots, n\}$ of consecutive integers from 1 to n is denoted by $[n]$. For a (un)directed graph G, we denote the vertex set and

[1] The $O^{\star}$ notation suppresses the polynomial dependence on the input size.

Voters	Preference Ranking over the Candidates									
v_1	6	3	7	8	5	1	2	4	9	10
v_2	6	1	7	4	9	5	8	3	10	2
v_3	2	10	8	9	1	5	7	6	3	4
v_4	2	10	4	5	3	8	7	1	6	9

Table 1: Example: Voting profile; v_1,v_2,v_3,v_4 are the voters, $\{1, 2, \ldots, 10\}$ is the set of candidates.

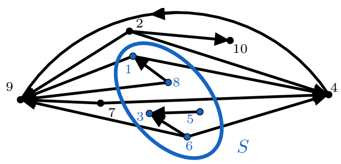

Figure 1: Example: The blue vertices in the set S is a weakly Gehrlein stable committee of size 5 for the voting profile given in Table 1.

the (edge) arc set of G by $V(G)$ and $\mathcal{A}(G)$, respectively. Let G be an undirected graph. We denote an edge between u and v as uv. Let G be a directed graph. We denote an arc from u to v by an ordered pair (u, v), and say that u is an in-neighbor of v and v is an out-neighbor of u. For $x \in V(G)$, $N^-_G(x) = \{y \in V(G) \colon (y, x) \in \mathcal{A}(G)\}$ and $N^+_G(x) = \{y \in V(G) \colon (x, y) \in \mathcal{A}(G)\}$. For $X \subseteq V(G)$, $G - X$ and $G[X]$ denote subgraphs of G induced on the vertex set $V(G) \setminus X$ and X, respectively. For $v_1, v_t \in V(G)$, a directed path from v_1 to v_t is denoted by $P = (v_1, v_2, \cdots, v_t)$, where $V(P) \subseteq V(G)$ and for each $i \in [t-1]$, $(v_i, v_{i+1}) \in \mathcal{A}(G)$. In a directed graph G, we say a vertex u is reachable from a vertex v, if there is directed path from v to u. Let $X \subseteq V(G)$. $G[X]$ is called a *strongly connected component* if every vertex in X is reachable from every other vertex in X. A strongly connected component, X, is called maximal if there does not exist a vertex $v \in V(G) \setminus X$ such that $X \cup \{v\}$ is also a strongly connected component. Let $x \in V(G)$. We define two sets $R^-_G(x)$ and $R^+_G(x)$ as follows. $R^-_G(x) = \{x\} \cup \{y \in V(G) \colon x \text{ is reachable from } y\}$ and $R^+_G(x) = \{x\} \cup \{y \in V(G) \colon y \text{ is reachable from } x\}$. We call $R^-_G(x)$ and $R^+_G(x)$ as *in-reachability* set and *out-reachability* set of x in G, respectively. For $S \subseteq V(G)$, $R^-_G(S) = \cup_{v \in S} R^-_G(v)$ and $R^+_G(S) = \cup_{v \in S} R^+_G(v)$. The subscript in the notation for the neighbourhood and the reachability sets may be omitted if the graph under consideration is clear from the context. Given an undirected graph G, complement of G is a graph G' such that $V(G') = V(G)$ and $E(G') = \{uv \colon u, v \in V(G) \text{ and } uv \notin E(G)\}$. For details on parameterized algorithms and kernelization, see [8, 18].

3 STRUCTURAL OBSERVATIONS

We start by making some simple observations that are crucial for most of our arguments. The proof of the next lemma follows from the definition of solution.

Lemma 3.1. *Let $(\mathcal{M}_\mathcal{E}, k)$ be a yes-instance of* GSCS*, and let S be a solution. Furthermore, let v_1 and v_t denote two vertices in $\mathcal{M}_\mathcal{E}$ such that there exists a path from v_1 to v_t in $\mathcal{M}_\mathcal{E}$. If $v_t \in S$, then $v_1 \in S$.*

As a corollary to Lemma 3.1, we get the following.

Corollary 3.1. *Let $(\mathcal{M}_\mathcal{E}, k)$ denotes a yes-instance of* GSCS*, and let S be a solution. Furthermore, let X denote a maximal strongly connected component in $\mathcal{M}_\mathcal{E}$.*

- *If $S \cap V(X) \neq \emptyset$, then $V(X) \subseteq S$*
- *If $v \in S$, then $R^-_{\mathcal{M}_\mathcal{E}}(v) \subseteq S$. Also, for every $v \in V(\mathcal{M}_\mathcal{E}) \setminus S$, $R^+_{\mathcal{M}_\mathcal{E}}(v) \cap S = \emptyset$.*

We also need the following hereditary property of the solution.

Lemma 3.2. *Let S be a solution of* GSCS *for $(\mathcal{M}_\mathcal{E}, k)$ and G be a subgraph of $\mathcal{M}_\mathcal{E}$. Then, $S' = S \cap V(G)$ is a solution of* GSCS *for $(G, |S'|)$.*

4 HARDNESS

In this section, we show that GSCS is W[1]-hard when parameterized by the solution size k. Towards this, we give a parameterized reduction from Multicolored Clique (MCQ), a well-known W[1]-hard problem [16], to GSCS running in polynomial time. MCQ is formally defined as follows.

> Multicolored Clique(MCQ) **Parameter:** k
> **Input:** A graph G, an integer k, and a partition of $V(G)$ into k sets $V_1, V_2, \cdots, V_k$ such that each V_i is an independent set.
> **Question:** Does there exist a set $Z \subseteq V(G)$ such that $G[Z]$ is a clique?

Given an instance of MCQ, we create an instance of GSCS as follows.

Construction. Let $(G, k, (V_1, \cdots, V_k))$ be an instance of MCQ. We construct the majority graph D in the following way (see Figure 2).

(1) For each vertex $v \in V(G)$, we introduce a vertex v and a directed cycle C_v passing through v of size k^2 in D. We call the vertex v as the node vertex, and vertices of C_v (including v) as the indicator vertices of v.
(2) For each $e \in E(G)$, we introduce a vertex w_e. We will refer to these vertices as edge vertices.
(3) For each edge $e \in E(G)$ with endpoints u and v, we introduce the arcs uw_e and vw_e in D.

We set the size of the solution to be $k' = k^3 + k(k-1)/2$. It is a well known fact that every directed graph is a majority graph of some election [14]. So D is a majority graph. This completes the description of the construction of an instance (D, k') of GSCS. Note that the steps of the reduction can be executed in polynomial time.

The intuitive idea of the reduction is the following. The construction enforces that when an edge vertex w_e is selected in a solution, both the endpoints of e are selected in the solution of GSCS. Moreover, when a node vertex v is selected, the directed cycle C_v is also selected in the solution. Intuitively, this indicates that v is in the solution to MCQ. The edge vertices in the solution of GSCS correspond to the edges that are in the solution of MCQ. We will show that due to size constraint of the solution there are exactly k^3 indicator vertices and $k(k-1)/2$ edge vertices.

Now, we formally prove the equivalence between the instance $(G, k, (V_1, \ldots, V_k))$ of MCQ and (D, k') of GSCS.

Correctness. We start by observing following property of D.

Observation 4.1. $V(D) = \biguplus_{v \in V(G)} C_v \uplus \biguplus_{e \in E(G)} w_e$.

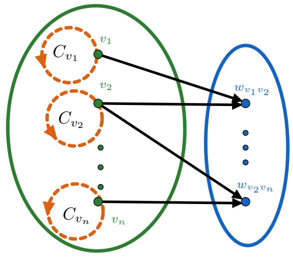

Figure 2: Construction of D. Here, $n = |V(G)|$, the green and blue vertices are the node vertices and edge vertices respectively, the vertices in the green set is the set of indicator vertices, and the orange dashed lines show the directed cycles of length k^2 in D.

Now for the correctness we show the following equivalence.

Lemma 4.1. $(G, k, (V_1, \cdots, V_k))$ *is a yes-instance of* MCQ *if and only if* $\mathcal{I} = (D, k')$ *is a yes-instance of* GSCS.

Proof. For the forward direction, assume that there exists $Z \subseteq V(G)$ such that $G[Z]$ is a clique in G with one vertex from each V_i, $i \in [k]$. We show that there exists a solution of (D, k'). Let $Z = \{v_1, v_2, \cdots, v_k\}$. Note that since $G[Z]$ is a clique, for each $\{i, j\} \subseteq [k]$, the edge $v_i v_j$ is present in G. We construct a set $S \subseteq V(D)$ from Z as follows.

$$S = \biguplus_{i \in [k]} V(C_{v_i}) \uplus \biguplus_{\{i,j\} \subseteq [k]} w_{v_i v_j}.$$

Now, we prove that S is a solution to (D, k'). Note that $|S| = k^3 + k(k-1)/2$. Therefore, it is sufficient to prove that there is no arc (x, y) in D such that $x \in V(D) \setminus S$ and $y \in S$. Equivalently, we prove that for each $y \in S$, $N^-(y) \subseteq S$, by considering the type of the vertex y.

If y is an indicator vertex, i.e., $y \in V(C_{v_i})$, for some $i \in [k]$, then, by the construction of D, y has exactly one in-neighbor which is in the cycle C_{v_i}. Hence, $N^-(y) \subseteq V(C_{v_i})$, i.e., $N^-(y) \subseteq S$. Suppose y is an edge vertex. Let $y = w_{v_i v_j}$. Then, the in-neighbors of y are v_i and v_j (see Figure 2). Notice that v_i, v_j are in the clique. Hence, both of them are in S. Therefore, for both the cases, $N^-(y) \subseteq S$. This proves the forward direction.

For the reverse direction, let S be a solution to (D, k'). Let $\mathcal{I}' = S \cap \uplus_{v \in V(G)} V(C_v)$, and $\mathcal{E}' = S \cap \uplus_{e \in E(G)} w_e$. Due to Observation 4.1, $\mathcal{I}'$ and $\mathcal{E}'$ are mutually disjoint. That is, $S = \mathcal{I}' \uplus \mathcal{E}'$. Also, note that since for each vertex $v \in V(G)$, C_v is a strongly connected component, by Corollary 3.1, if $V(C_v) \cap S \neq \emptyset$, then $V(C_v) \subseteq S$.

Now we define $V^\star = \{v \in V(G) \colon V(C_v) \subseteq \mathcal{I}'\}$ and $E^\star = \{e \in E(G) \colon w_e \in \mathcal{E}'\}$. We will prove that $G' = (V^\star, E^\star)$ is a solution to MCQ to (G, k).

Claim 4.1. $|V^\star| = k$.

Proof. Suppose $|V^\star| = k^\star < k$. For each $v \in V^\star$, $V(C_v) \subseteq \mathcal{I}'$. Hence, the number of indicator vertices in S is $k^\star k^2$, which is less than k^3. Since $k' = k^3 + k(k-1)/2$, there must be strictly more than $k(k-1)/2$ edge vertices in S. However, since each edge vertex has two node vertices as in-neighbors, and there are $k^\star$ node vertices in S, there are at most $\binom{k^\star}{2}$ edge vertices in S. So the number of vertices in S is $k^\star k^2 + \binom{k^\star}{2}$. This contradicts that the size of S is $k' = k^3 + k(k-1)/2$.

Now, suppose $|V^\star| = k^\star > k$. Then, there are at least $k^\star k^2$ vertices in S. Since $k^\star k^2 \geq (k+1)k^2$, we have that $|S| > k'$, a contradiction. □

Claim 4.2. $|E^\star| = k(k-1)/2$.

Proof. From Claim 4.1, there are k node vertices in S. Therefore, S has k^3 many indicator vertices. From Observation 4.1, we know that the remaining vertices of S are from the set of edge vertices. Since $|S| = k^3 + k(k-1)/2$, there are $k(k-1)/2$ edge vertices. □

Now, we prove that the vertices are consistent with the edges. That is, we prove that if edge $uv \in E^\star$, then $u, v \in V^\star$. Note that if $w_{uv} \in S$, then since u, v are the in-neighbors of the edge vertex w_{uv}, we have that $u, v \in S$. Hence, $V(C_u) \subseteq S$ and $V(C_v) \subseteq S$. Therefore, if $uv \in E^\star$, then $u, v \in V^\star$. Moreover, since $G' = (V^\star, E^\star)$, $|E^\star| = k(k-1)/2$, and $|V^\star| = k$, we can infer that G' is a clique of size k. Since each V_i is an independent set and G' is a clique, G' cannot contain two vertices from any V_i. Hence, G' is a solution to MCQ. □

Hence, we have proved the following theorem.

Theorem 4.2. GSCS *is* W[1]*-hard when parameterized by the size of solution.*

5 EXACT ALGORITHMS FOR GEHRLEIN STABLE COMMITTEE SELECTION

Let $(\mathcal{M}_{\mathscr{E}} = (C, \mathcal{A}), k)$ be an instance of GSCS. Furthermore, let n denote the number of candidates or the number of vertices in $\mathcal{M}_{\mathscr{E}}$. Observe that we can design an algorithm for GSCS by enumerating all vertex subsets of size k and checking whether it forms a solution. This algorithm runs in time $O^\star(\binom{n}{k}) = O^\star(2^n)$. So a natural question is whether we can design an exact algorithm that improves over this brute-force enumeration algorithm.

In this section, we design a non-trivial exact algorithm for GSCS running in time $O^\star(1.2207^n)$. The main idea of the algorithm is inspired by Corollary 3.1. We find a subset of vertices with the property that either all of them go to the solution or none of them go to the solution. Once we have identified such a subset we recursively solve two subproblems, one where these vertices are part of a solution we are constructing, and the other where none of these vertices are part of the solution. Observe that a maximal strongly connected component provides a natural candidate of subset vertices. The algorithm indeed branches on strongly connected component of sufficiently large size and when we do not have a maximal strongly connected component of sufficiently big size we solve the problem in polynomial time. We first give the polynomial time subcase of our problem and then design the promised exact algorithm.

5.1 A polynomial time subcase

In this section, we give a polynomial time algorithm for GSCS when majority graph, $\mathcal{M}_{\mathscr{E}}$, is a disjoint union of directed acyclic graphs and strongly connected components. That is, if we look at the connected components of underlying undirected graph of majority graph (that is, consider majority graph without the edge orientations), then they are either a directed acyclic graph or a strongly connected component in majority graph. We denote such a family of graphs by $\mathscr{F}_{\mathbf{dag+scc}}$. Furthermore, let $\mathscr{F}_{\mathbf{scc}}$ denote the family of disjoint union of strongly connected components, and $\mathscr{F}_{\mathbf{dag}}$ denote the family of disjoint union of directed acyclic graphs.

We first give algorithms for GSCS on $\mathscr{F}_{\mathbf{scc}}$ and $\mathscr{F}_{\mathbf{dag}}$, and then use these to give our algorithm on $\mathscr{F}_{\mathbf{dag+scc}}$. We design an algorithm for GSCS on $\mathscr{F}_{\mathbf{scc}}$, by reducing it to the well-known Subset Sum problem. In the Subset Sum problem, given a set of integers $X = \{x_1, \cdots, x_n\}$, and an integer W, the goal is to find a set $X' \subseteq X$ such that $\sum_{x_i \in X'} x_i = W$.

Lemma 5.1. GSCS *on* $\mathscr{F}_{\mathbf{scc}}$ *can be reduced to* Subset Sum *in* $O(n)$ *time.*

Lemma 5.2. [24] *Given an instance* (X, W) *of* Subset Sum*, there exists an algorithm that solves* Subset Sum *in* $O(nW)$ *time, where* $n = |X|$.

Using Lemmas 5.1 and 5.2, we get the following result.

Lemma 5.3. GSCS *can be solved in* $O(nk)$ *time on* $\mathscr{F}_{\mathbf{scc}}$*. Here, n is number of vertices in the input graph, and k is the size of solution.*

Now, we give a polynomial time algorithm for GSCS on $\mathscr{F}_{\mathbf{dag}}$. The algorithm just selects the first k vertices of the topological ordering in the solution.

Lemma 5.4. GSCS *can be solved in* $O(n + m)$ *time on* $\mathscr{F}_{\mathbf{dag}}$*. Here, n is the number of vertices in the input graph, and m is the number of arcs in the graph.*

Now, we are ready to give a polynomial time algorithm for GSCS on $\mathscr{F}_{\mathbf{dag+scc}}$. The algorithm first guesses how many vertices a solution contains from strongly connected components and **dags**, respectively. Then, it runs the algorithms given in Lemmas 5.3 and 5.4 and compute the desired solution.

Theorem 5.5. GSCS *can be solved in* $O(nk^2)$ *time on* $\mathscr{F}_{\mathbf{dag+scc}}$*. Here, n is number of vertices in the input graph, and k is the size of solution.*

5.2 Exact exponential time algorithm

Now, we proceed towards presenting the exact exponential algorithm for GSCS. Towards this, we first prove the following structural result.

Lemma 5.6. *Let* $(\mathcal{M}_{\mathscr{E}}, k)$ *be an instance of* GSCS *such that* $\mathcal{M}_{\mathscr{E}} \notin \mathscr{F}_{\mathbf{dag+scc}}$*. Then,* $\mathcal{M}_{\mathscr{E}}$ *has a strongly connected component X of size at least three such that either* $|R^-(X)| \geq 4$ *or* $|R^+(X)| \geq 4$.

Proof. If there does not exist a strongly connected component of $\mathcal{M}_{\mathscr{E}}$ of size at least three, then since $\mathcal{M}_{\mathscr{E}}$ does not contain parallel edges and self loops, $\mathcal{M}_{\mathscr{E}}$ is a **dag**, a contradiction to the fact that $\mathcal{M}_{\mathscr{E}}$ does not belong to $\mathscr{F}_{\mathbf{dag+scc}}$. Thus, we know that there exists a strongly connected component of size at least 3.

Let X be a maximal strongly connected component of $\mathcal{M}_{\mathscr{E}}$ such that $|X|$ is maximized. If $|X| \geq 4$, we are done. Furthermore, observe that if there exists a maximal strongly connected component X, such that $|X| = 3$, and either $|R^-(X)| \geq 4$ or $|R^+(X)| \geq 4$, we are done. This implies that every maximal strongly connected component of size 3 is a connected component in itself in the underlying undirected graph of $\mathcal{M}_{\mathscr{E}}$. If we remove these components from $\mathcal{M}_{\mathscr{E}}$ we get a directed graph that does not have any strongly connected component and hence it is a **dag**. This implies that $\mathcal{M}_{\mathscr{E}}$ belongs to $\mathscr{F}_{\mathbf{dag+scc}}$, a contradiction. This concludes the proof. ◻

Now, we are ready to present the algorithm. Let $(\mathcal{M}_{\mathscr{E}}, k)$ be an instance of GSCS. We apply the following branching rule exhaustively.

Branching Rule 5.1. *Suppose X is a strongly connected component in* $\mathcal{M}_{\mathscr{E}}$ *such that* $|X| \geq 3$ *and either* $|R^-(X)| \geq 4$ *or* $|R^+(X)| \geq 4$*. Branch by adding* $R^-(X)$ *to the solution or deleting* $R^+(X)$ *from* $\mathcal{M}_{\mathscr{E}}$*. Recurse on the instances* $(\mathcal{M}_{\mathscr{E}} - R^-(X), k - |R^-(X)|)$ *and* $(\mathcal{M}_{\mathscr{E}} - R^+(X), k)$*, respectively.*

Lemma 5.7. *Branching Rule 5.1 is correct.*

Proof. We claim that $(\mathcal{M}_{\mathscr{E}}, k)$ is a yes-instance of GSCS if and only if either $(\mathcal{M}_{\mathscr{E}} - R^-(X), k - |R^-(X)|)$ is a yes-instance of GSCS or $(\mathcal{M}_{\mathscr{E}} - R^+(X), k)$ is a yes-instance of GSCS. In the forward direction, let $(\mathcal{M}_{\mathscr{E}}, k)$ be a yes-instance of GSCS and S be one of its solutions. Consider a strongly connected component, X, of $\mathcal{M}_{\mathscr{E}}$. If $x \in S \cap X$, then by Corollary 3.1, we have that $R^-(X) \subseteq S$. Using Lemma 3.2, $S \setminus R^-(X) = S \cap V(\mathcal{M}_{\mathscr{E}} - R^-(X))$ is a solution of GSCS for $(\mathcal{M}_{\mathscr{E}} - R^-(X), k - |R^-(X)|)$. Now, suppose $X \cap S = \emptyset$. By Corollary 3.1, $R^+(X) \cap S = \emptyset$. Therefore, by Lemma 3.2, S is also a solution to $(\mathcal{M}_{\mathscr{E}} - R^+(X), k)$. This completes the proof in the forward direction.

In the backward direction, let S be a solution to GSCS for $(\mathcal{M}_{\mathscr{E}} - R^-(X), k - |R^-(X)|)$. We claim that $S \cup R^-(X)$ is a solution to $(\mathcal{M}_{\mathscr{E}}, k)$. Suppose not, then there exists $u \in S$ and $v \in V(\mathcal{M}_{\mathscr{E}}) \setminus S$ such that $(v, u) \in \mathcal{A}(\mathcal{M}_{\mathscr{E}})$. If $u \notin R^-(X)$, then (v, u) also belongs to $\mathcal{A}(\mathcal{M}_{\mathscr{E}} - R^-(X))$, a contradiction to the fact that S is a solution to $(\mathcal{M}_{\mathscr{E}} - R^-(X), k - |R^-(X)|)$. Now, suppose $u \in R^-(X)$. Since $v \notin S$, $v \notin R^-(X)$, a contradiction to the fact that $(v, u) \in \mathcal{A}(\mathcal{M}_{\mathscr{E}})$. This proves that $S \cup R^-(X)$ is a solution to $(\mathcal{M}_{\mathscr{E}}, k)$. Now suppose that S is a solution to GSCS for $(\mathcal{M}_{\mathscr{E}} - R^+(X), k)$. We claim that S is also a solution to $(\mathcal{M}_{\mathscr{E}}, k)$. Suppose not, then there exists $u \in S$ and $v \in V(\mathcal{M}_{\mathscr{E}}) \setminus S$ such that $(v, u) \in \mathcal{A}(\mathcal{M}_{\mathscr{E}})$. If $v \notin R^+(X)$, then (v, u) also belongs to $\mathcal{A}(\mathcal{M}_{\mathscr{E}} - R^+(X))$, a contradiction to that S is a solution to $(\mathcal{M}_{\mathscr{E}} - R^+(X), k)$. Now, suppose $v \in R^+(X)$. Since $u \in S$, $u \notin R^+(X)$, a contradiction to that $(v, u) \in \mathcal{A}(\mathcal{M}_{\mathscr{E}})$. ◻

Theorem 5.8. GSCS *can be solved in* $O^\star(1.2207^n)$ *time optimally, where n is the number of vertices in* $\mathcal{M}_{\mathscr{E}}$.

Proof. Given an instance $(\mathcal{M}_{\mathscr{E}}, k)$ of GSCS, we first check whether $\mathcal{M}_{\mathscr{E}}$ belongs to the family $\mathscr{F}_{\mathbf{dag+scc}}$. If yes, then we can solve the problem in polynomial time using Theorem 5.5. Otherwise, using Lemma 5.6 there exists a strongly connected component, X of size at least three such that either $|R^-(X)| \geq 4$ or $|R^+(X)| \geq 4$. Now, we apply Branching Rule 5.1. The safeness of algorithm follows from the safeness of branching rule. The running time of the algorithm is governed by the recurrence, $T(n) \leq T(n-3) + T(n-4)$, which solves to $O^\star(1.2207^n)$. ◻

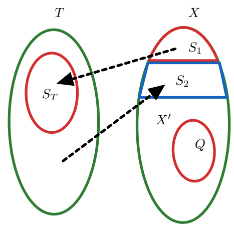

Figure 3: An illustration of Algorithm 1 where vertices in the red sets are in the solution

6 FPT ALGORITHMS FOR GSCS

Given an instance $(\mathcal{M}_{\mathcal{E}}, k)$ of GSCS, let q be the size of **tvd** of $\mathcal{M}_{\mathcal{E}}$ and $l = \binom{n}{2} - |\mathcal{A}(\mathcal{M}_{\mathcal{E}})|$. In this section, we design fixed parameter algorithms for GSCS with respect to parameters q and l.

We first give an FPT algorithm (Algorithm 1) for GSCS when parameterized by q. First, we state a known result which is the starting point of our algorithm.

Proposition 6.1. [1] GSCS *can be solved in the polynomial time if the majority graph* $\mathcal{M}_{\mathcal{E}}$ *is a tournament. Moreover, such a solution is unique, if exists.*

Let X be a *tvd* of $\mathcal{M}_{\mathcal{E}}$. Note that X is a vertex cover - a set of vertices such that each edge is incident to at least one vertex of the set - of the complement graph of the underlying undirected graph of $\mathcal{M}_{\mathcal{E}}$. Hence, it can be computed in $O^\star(1.2738^q)$ time, where $q = |X|$, using the FPT algorithm proposed by Chen et al. [5]. Note that $T = \mathcal{M}_{\mathcal{E}} - X$ is a tournament. Proposition 6.1 says that every tournament has a unique solution. Thus, for our algorithm we first guess how many vertices from T are present in our potential solution to $(\mathcal{M}_{\mathcal{E}}, k)$. Once, we have guessed this number k_1, we run the algorithm mentioned in Proposition 6.1 and find the unique solution of size k_1, say S_T, if exists. Having found the set S_T, we know that no vertex of $T - S_T$ goes into the solution. Hence, we now apply Corollary 3.1 and reduce the problem to a directed graph induced on a subset of X. At this stage we run the exact exponential time algorithm described in Theorem 5.8 and we are done. See Figure 3 for illustration of the algorithm. A detailed description of the algorithm is given in Algorithm 1.
Now, we prove the correctness of this algorithm.

Lemma 6.2. *Algorithm 1 is correct.*

Proof. To prove the correctness of algorithm, we prove that if Algorithm 1 returns a non-empty set S, then it is a solution of GSCS to $(\mathcal{M}_{\mathcal{E}}, k)$, otherwise $(\mathcal{M}_{\mathcal{E}}, k)$ does not have a solution of GSCS.

Case A: Suppose $S \neq \emptyset$. We claim that S is a solution to $(\mathcal{M}_{\mathcal{E}}, k)$. Suppose not, then either $|S| \neq k$ or there exists $w \in S$, and $w' \in V(\mathcal{M}_{\mathcal{E}}) \setminus S$ such that $(w', w) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$. By the construction of S, if $S \neq \emptyset$, then $|S| = k$. Now, suppose that there exists $w \in S$ and $w' \in V(\mathcal{M}_{\mathcal{E}}) \setminus S$ such that $(w', w) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$. Following four cases are possible.

Case 1: Suppose $w, w' \in V(T)$. Since $S \cap V(T) = S_T$, $w \in S_T$ and $w' \notin S_T$. Since T is an induced subgraph of $\mathcal{M}_{\mathcal{E}}$, $(w', w) \in \mathcal{A}(T)$, this contradicts the fact that S_T is a solution to $(T, |S_T|)$.

Algorithm 1: FPT for GSCS

Input: A majority graph $\mathcal{M}_{\mathcal{E}}$, a **tvd**, X of $\mathcal{M}_{\mathcal{E}}$, and an integer k.
Output: S, which is a solution of GSCS for $(\mathcal{M}_{\mathcal{E}}, k)$, if non-empty

```
S_1 = ∅, Q = ∅;
for each k_1 ∈ [k] do
    if T has a solution of GSCS of size k_1 then
        let S_T be the solution of (T, k_1) computed using Proposition 6.1 ;
        S_1 = R^-_{M_E}(S_T) ∩ X, S_2 = R^+_{M_E}(V(T) \ S_T) ∩ X;
        if S_1 ∩ S_2 = ∅ then
            X' = X \ (S_1 ∪ S_2), k_2 = k − |S_T ⊎ S_1|;
            if M_E[X'] has a solution of GSCS of size k_2 then
                let Q be a solution of (M_E[X'], k_2) computed using Theorem 5.8;
                S = S_T ⊎ S_1 ⊎ Q;
                return S.
end
return S = ∅
```

Case 2: Suppose $w, w' \in X$. Note that by the construction of X', $X \setminus X' = S_1 \uplus S_2$. Note that $S_1 \subseteq S$ and $S_2 \cap S = \emptyset$. Following four cases are possible.

Case(i): Suppose $w, w' \in X \setminus X'$. Since $w \in S$, it follows that $w \in S_1$. Since $w' \notin S$, $w' \in S_2$. Since $(w', w) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$, and w is in the *in-reachability* set of S_T in $\mathcal{M}_{\mathcal{E}}$, w' also belongs to *in-reachability* set of S_T in $\mathcal{M}_{\mathcal{E}}$. Hence, $w' \in S_1$, a contradiction to that S_1 and S_2 are disjoint.

Case(ii): Suppose $w \in X \setminus X'$ and $w' \in X'$. Since $w \in S$, $w \in S_1$. As argued above, since $(w', w) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$, w' also belongs to S_1, a contradiction to that X' and S_1 are disjoint.

Case(iii): Suppose $w \in X'$ and $w' \in X \setminus X'$. Since $w' \notin S$, we have that $w' \in S_2$. Since $(w', w) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$, and w' is in *out-reachability* set of $V(T) \setminus S_T$ in $\mathcal{M}_{\mathcal{E}}$, w also belongs to *out-reachability* set of $V(T) \setminus S_T$ in $\mathcal{M}_{\mathcal{E}}$ and hence $w \in S_2$, a contradiction to that X' and S_2 are disjoint.

Case(iv): Suppose $w, w' \in X'$. Since $Q \subseteq S$, $w \in Q$ and $w' \notin Q$. Since $\mathcal{M}_{\mathcal{E}}[X']$ is an induced subgraph of $\mathcal{M}_{\mathcal{E}}$, we have that $(w', w) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}}[X'])$, this contradicts that Q is a solution to $(\mathcal{M}_{\mathcal{E}}[X'], |Q|)$.

Case 3: Suppose $w \in V(T)$ and $w' \in X$. Since $w \in S$, $w \in S_T$. Since $w' \notin S$, there are two cases, either $w' \in S_2$ or $w' \in X' \setminus Q$. Since $(w', w) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$ and $w \in S_T$, w' belongs to the *in-reachability* set of S_T in $\mathcal{M}_{\mathcal{E}}$ and hence $w' \in S_1$. If $w' \in S_2$, then it contradicts that S_1 and S_2 are disjoint. If $w' \in X'$, then it contradicts that S_1 and X' are disjoint.

Case 4: Suppose $w \in X$ and $w' \in V(T)$. Since $w \in S$, w either belongs to S_1 or Q. Since $(w', w) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$ and

$w' \in V(T) \setminus S$, clearly, w belongs to the *out-reachability* set of $V(T) \setminus S$ in $\mathcal{M}_{\mathcal{E}}$. Therefore, $w \in S_2$. If $w \in S_1$, then it contradicts that S_1 and S_2 are disjoint. If $w \in Q$, then it contradicts that Q and S_2 are disjoint.

Case B: Suppose $S = \emptyset$. We claim that $(\mathcal{M}_{\mathcal{E}}, k)$ does not have a solution of GSCS. Towards the contrary, let Z be a solution to $(\mathcal{M}_{\mathcal{E}}, k)$. Let $Z_T = V(T) \cap Z$ and $k' = |Z_T|$. Using Lemma 3.2, Z_T is a solution to (T, k'). Since T is a tournament, by uniqueness of solution of tournament (Proposition 6.1), $S_T = Z_T$, where S_T is a set returned in Step 4 of Algorithm 1 when $k_1 = k'$. Let $Z_1 = R^-_{\mathcal{M}_{\mathcal{E}}}(Z_T) \cap X$ and $Z_2 = R^+_{\mathcal{M}_{\mathcal{E}}}(V(T) \setminus Z_T) \cap X$. Using Corollary 3.1, $Z_1 \subseteq Z$ and $Z_2 \cap Z = \emptyset$. Therefore, Z_1 and Z_2 are disjoint. Clearly, $S_1 = Z_1$ and $S_2 = Z_2$ in Step 5 of Algorithm 1. Let $X' = X \setminus (Z_1 \cup Z_2)$. Note that $X = X' \uplus Z_1 \uplus Z_2$. Since $Z1 \subseteq Z$ and $Z_2 \cap Z = \emptyset$, $Z \cap X = Z_1 \uplus (Z \cap X')$. Using Lemma 3.2, $Z' = Z \cap X'$ is a solution to $(\mathcal{M}_{\mathcal{E}}[X'], |Z'|)$. Since there exist a solution to $(\mathcal{M}_{\mathcal{E}}[X'], |Z'|)$, algorithm finds a solution Q to $(\mathcal{M}_{\mathcal{E}}[X'], |Z'|)$ in Step 9. Since $Z = Z_T \uplus (Z \cap X)$ and $Z \cap X = Z_1 \uplus Z'$, $|Z'| = k - |Z_T \uplus Z_1| = k - |S_T \uplus S_1|$. Therefore, Algorithm 1 returns $S = S_T \uplus S_1 \uplus Q$, a contradiction to that $S = \emptyset$.

□

Lemma 6.3. *The running time of Algorithm 1 is $O^\star(1.2207^q)$, where q is the size of* **tvd** *of majority graph $\mathcal{M}_{\mathcal{E}}$.*

Proof. In the algorithm, set S_T (Step 4) can be computed in polynomial time using Proposition 6.1 and set Q (Step 9) can be obtained using Theorem 5.8 in $O^\star(1.2207^q)$ time. Hence, the running time of the algorithm is $O^\star(1.2207^q)$. □

Theorem 6.4. GSCS *can be solved in $O^\star(1.2738^q)$ time, where q is the size of* **tvd** *of majority graph $\mathcal{M}_{\mathcal{E}}$.*

Proof. Given an instance $(\mathcal{M}_{\mathcal{E}}, k)$ of GSCS, we first compute vertex cover, X, of the complement graph of underlying undirected graph of $\mathcal{M}_{\mathcal{E}}$ using an FPT algorithm which runs in $O^\star(1.2738^q)$ time, where q is the size of vertex cover [5]. Now using Algorithm 1, we compute a solution S of GSCS to $(\mathcal{M}_{\mathcal{E}}, k)$, if exists. The correctness of algorithm follows from Lemma 6.2. Since using Lemma 6.3, the running time of Algorithm 1 is $O^\star(1.2207^q)$, and we compute X in $O^\star(1.2738^q)$ time, it follows that GSCS can be solved in $O^\star(1.2738^q)$ time. □

Now, we give an FPT algorithm for GSCS when the number of pairs of candidates which are tied among each other in the pairwise majority contest are bounded.

Theorem 6.5. GSCS *can be solved in $O^\star(1.2207^l)$ time when the number of missing arcs in majority graph is l.*

Proof. Given an instance $(\mathcal{M}_{\mathcal{E}}, k)$ of GSCS, let X be a set of vertices obtained by adding a vertex from each of the missing arcs. Note that X is a **tvd** of $\mathcal{M}_{\mathcal{E}}$, and $|X| \leq l$. Now, using Algorithm 1, we compute a solution S of GSCS to $(\mathcal{M}_{\mathcal{E}}, k)$, if exists. The correctness of algorithm follows from Lemma 6.2. Since $|X| \leq l$, using Lemma 6.3, we can solve GSCS in $O^\star(1.2207^l)$ time. □

7 A LINEAR VERTEX KERNEL FOR GSCS

In this section, we show that GSCS admits a kernel with $O(q)$ vertices, where q is the size of **tvd** of $\mathcal{M}_{\mathcal{E}}$. That is, we give a polynomial time algorithm that given an instance $(\mathcal{M}_{\mathcal{E}}, k)$ of GSCS returns an instance $(\mathcal{M}'_{\mathcal{E}}, k')$ of GSCS such that $(\mathcal{M}_{\mathcal{E}}, k)$ is a yes-instance of GSCS if and only if $(\mathcal{M}'_{\mathcal{E}}, k')$ is a yes-instance of GSCS and $|V(\mathcal{M}'_{\mathcal{E}})| \leq 4q + 1$.

Let $(\mathcal{M}_{\mathcal{E}}, k)$ be an instance of GSCS. Let X be a set such that $T = \mathcal{M}_{\mathcal{E}} - X$ is a tournament. Let $t = |V(T)|$. We know that every tournament T has a Hamiltonian path—a path that visits every vertex exactly once. Furthermore, a Hamiltonian path in tournament can be computed in polynomial time [20]. Let $\mathcal{H} = (v_1, v_2, \cdots, v_t)$ be one such path. Now notice that no vertex in $\{v_{k+1}, \cdots, v_t\}$ belongs to any solution to $(\mathcal{M}_{\mathcal{E}}, k)$ and thus, we should be able to find a reduction rule that can reduce the size of T to $k + 1$. However, this is still not the desired kernel. Next, we change our perspective and see which vertices from T must be in a solution of size k. Once we detect such a vertex, we can use Corollary 3.1 to find a desired reduction rule.

Before diving into the details of the algorithm, we give an alternate polynomial time algorithm to find a solution of GSCS, when the majority graph is a tournament. This will be crucially used in designing the kernelization algorithm.

Lemma 7.1. *Let (G, k) be an instance of* GSCS*, where G is a tournament. Let $|V(G)| = t$, and $\mathcal{H} = (v_1, v_2, \cdots, v_t)$ be a Hamiltonian path in G. Furthermore, let $S = \{v_1, v_2, \cdots, v_k\}$. If $|R^-_G(S)| = k$, then S is a solution of* GSCS *to (G, k). Moreover, it is the unique solution and can be computed in polynomial time.*

Proof. Since $|R^-_G(S)| = k$, for every $v \in S$, $N^-_G(v) \subseteq S$. Hence, S is a solution of GSCS to (G, k). Next, we will prove that it is the unique solution. Suppose not, then let $S' (\neq S)$ be a solution of GSCS to (G, k). Since $|S'| = |S|$, there exists a vertex $v^\star \in S' \setminus S$, i.e., $v^\star \in \{v_{k+1}, \cdots, v_t\}$. Note that $P = (v_1, v_2, \cdots, v^\star)$ is a subpath of $\mathcal{H}$. Each vertex in P can reach $v^\star$ via the path P. Hence, $V(P) \subseteq R^-_G(v^\star)$. Also, since $v^\star \notin \{v_1, v_2, \cdots, v_k\}$, the number of vertices in P is at least $k + 1$. Hence, $|R^-_G(v^\star)| \geq k + 1$. Since $v^\star \in S'$, using Corollary 1, $R^-_G(v^\star) \subseteq S'$, this contradicts that S' is a solution to (G, k). Since the Hamiltonian path in a tournament can be found in polynomial time [21], S can be computed in polynomial time. □

Now, we are ready to give the detailed description of the algorithm. First, we describe how to find a **tvd**, X, of size at most $2q$. Recall that every **tvd** of $\mathcal{M}_{\mathcal{E}}$ is also a vertex cover of the complement graph, $\overline{G}$, of the underlying undirected graph of $\mathcal{M}_{\mathcal{E}}$. Thus, to get the desired X, we find a vertex cover of $\overline{G}$ using a well-known factor 2-approximation algorithm for the Vertex Cover problem [3]. Note that $T = \mathcal{M}_{\mathcal{E}} - X$ is a tournament. Next, we define a sequence of reduction rules. At any point of time we apply the lowest indexed reduction rule that is applicable. That is, a rule is applied only when none of the preceding rules can be applied. After an application of any rule, we reuse the notation $\mathcal{M}_{\mathcal{E}}$ to denote the reduced majority graph.

Reduction Rule 7.1. *Let $(\mathcal{M}_{\mathcal{E}}, k)$ be an instance of* GSCS *and let $T = \mathcal{M}_{\mathcal{E}} - X$. Furthermore, let $\mathcal{H} = (v_1, v_2, \cdots, v_t)$ be a Hamiltonian path in T. If $t > k+1$, then construct the majority graph $\mathcal{M}'_{\mathcal{E}}$ such that*

$\mathcal{M}'_{\mathcal{E}} = \mathcal{M}_{\mathcal{E}} - \{v_t\}$, *and* $N^+_{\mathcal{M}'_{\mathcal{E}}}(v_{t-1}) = N^+_{\mathcal{M}_{\mathcal{E}}}(v_{t-1}) \cup N^+_{\mathcal{M}_{\mathcal{E}}}(v_t) \setminus \{v_{t-1}\}$. *The resulting instance is* $(\mathcal{M}'_{\mathcal{E}}, k)$.

Lemma 7.2. *Reduction Rule 7.1 is safe.*

Proof. Suppose $(\mathcal{M}_{\mathcal{E}}, k)$ is a yes-instance of GSCS and S is a solution to $(\mathcal{M}_{\mathcal{E}}, k)$. We prove that S is also a solution to $(\mathcal{M}'_{\mathcal{E}}, k)$. Suppose not, then there exists $x \in S$, and $y \in V(\mathcal{M}'_{\mathcal{E}}) \setminus S$ such that $(y, x) \in \mathcal{A}(\mathcal{M}'_{\mathcal{E}})$. If $y \neq v_{t-1}$, then (y, x) also belongs to $\mathcal{A}(\mathcal{M}_{\mathcal{E}})$, a contradiction to the fact that S is a solution to $(\mathcal{M}_{\mathcal{E}}, k)$. Suppose $y = v_{t-1}$, then by the construction of $\mathcal{M}'_{\mathcal{E}}$, either $(v_{t-1}, x) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$ or $(v_t, x) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$. If $(v_{t-1}, x) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$, then it contradicts the fact that S is a solution to $(\mathcal{M}_{\mathcal{E}}, k)$. Suppose $(v_t, x) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$. Now, we show that $v_t \notin S$ which will lead to the contradiction that S is a solution to $(\mathcal{M}_{\mathcal{E}}, k)$. By Lemma 3.2, $S \cap V(T)$ is a solution to $(T, |S \cap V(T)|)$ and by Lemma 7.1, we know that it is the unique solution to $(T, |S \cap V(T)|)$. Since $t > k + 1$, by Lemma 7.1, we have that $v_t \notin S \cap V(T)$. Since $v_t \in V(T)$, it follows that $v_t \notin S$.

For the other direction, suppose S' is a solution to $(\mathcal{M}'_{\mathcal{E}}, k)$. We prove that S' is also a solution to $(\mathcal{M}_{\mathcal{E}}, k)$. Suppose not, then there exists $x \in S'$, and $y \in V(\mathcal{M}_{\mathcal{E}}) \setminus S'$ such that $(y, x) \in \mathcal{A}(\mathcal{M}_{\mathcal{E}})$. If $y \neq v_t$, then (y, x) also belongs to $\mathcal{A}(\mathcal{M}'_{\mathcal{E}})$, a contradiction to that S' is a solution to $(\mathcal{M}'_{\mathcal{E}}, k)$. Suppose $y = v_t$. Since $T' = T - \{v_t\}$ is a tournament, using Lemmas 3.2, and 7.1, we have that $S' \cap V(T')$ is the unique solution to $(T', |S' \cap V(T')|)$. Note that $(v_1, \cdots, v_{t-1})$ is also a Hamiltonian path in T'. Since $t > k + 1$, by Lemma 7.1, we have that $v_{t-1} \notin S' \cap V(T')$. Hence, $v_{t-1} \notin S'$. Therefore, $x \neq v_{t-1}$. Since $y = v_t$, and $(y, x) \in \mathcal{A}(\mathcal{M}'_{\mathcal{E}})$, by construction of $\mathcal{M}'_{\mathcal{E}}$, we have that $(v_{t-1}, x) \in \mathcal{A}(\mathcal{M}'_{\mathcal{E}})$. Now, since $v_{t-1} \notin S'$, it contradicts that S' is a solution to $(\mathcal{M}'_{\mathcal{E}}, k)$. □

If Reduction Rule 7.1 is not applicable, then $|V(T)| \leq k + 1$. Since $|X| \leq 2q$, we have that $\mathcal{M}_{\mathcal{E}}$ has at most $2q + k + 1$ vertices. Hence, the next lemma follows.

Lemma 7.3. *If* $k \leq 2q$*, and Reduction Rule 7.1 is not applicable, then* $|V(\mathcal{M}_{\mathcal{E}})| \leq 4q + 1$.

Now, it remains to bound the number of vertices in T by $\mathcal{O}(q)$ when $k > 2q$.

Reduction Rule 7.2. *Let* $(\mathcal{M}_{\mathcal{E}}, k)$ *be an instance of* GSCS *and let* $T = \mathcal{M}_{\mathcal{E}} - X$. *And, let* $\mathcal{H} = (v_1, v_2, \cdots, v_{|V(T)|})$ *be a Hamiltonian path in* T. *If* $k > 2q$ *and* $|R^-_T(\{v_1, \cdots, v_{k-2q}\})| > k$*, then output a* n*o instance of constant size.*

Lemma 7.4. *Reduction Rule 7.2 is safe.*

Proof. Suppose $k > 2q$, and $|R^-(\{v_1, \cdots, v_{k-2q}\})| > k$. We prove that $(\mathcal{M}_{\mathcal{E}}, k)$ is a no instance of GSCS. Suppose not, let S be a solution of GSCS to $(\mathcal{M}_{\mathcal{E}}, k)$. Since $k > 2q$, and $|X| \leq 2q$, we have that any k size solution for $\mathcal{M}_{\mathcal{E}}$ must contain vertices outside X. That is, it must contain at least $k - 2q$ vertices of T. Note that using Lemma 3.2, $S \cap V(T)$ is a solution for $(T, |S \cap V(T)|)$ and using Lemma 7.1, it is the unique solution for $(T, |S \cap V(T)|)$. Since $|S \cap V(T)| \geq k - 2q$, using Lemma 7.1, $\{v_1, \cdots, v_{k-2q}\} \subseteq S \cap V(T)$. Hence $\{v_1, \cdots, v_{k-2q}\} \subseteq S$. Since $R^-_T(\{v_1, \cdots, v_{k-2q}\}) > k$, $|S| > k$, a contradiction to that S is a solution to $(\mathcal{M}_{\mathcal{E}}, k)$. □

Reduction Rule 7.3. *Let* $(\mathcal{M}_{\mathcal{E}}, k)$ *be an instance of* GSCS *and let* $T = \mathcal{M}_{\mathcal{E}} - X$. *Furthermore, let* $\mathcal{H} = (v_1, v_2, \cdots, v_t)$ *be a Hamiltonian path in* T. *If* $k > 2q$*, then delete* $R^-_{\mathcal{M}_{\mathcal{E}}}(v_1)$ *from* $\mathcal{M}_{\mathcal{E}}$. *The reduced instance is* $(\mathcal{M}'_{\mathcal{E}}, k')$*, where* $\mathcal{M}'_{\mathcal{E}} = \mathcal{M}_{\mathcal{E}} - R^-_{\mathcal{M}_{\mathcal{E}}}(v_1)$ *and* $k' = k - |R^-_{\mathcal{M}_{\mathcal{E}}}(v_1)|$.

Safeness of Reduction Rule 7.3 follows from Corollary 3.1 and Lemma 7.1. Now, we give the main result of this section.

Theorem 7.5. GSCS *admits a kernel with* $4q + 1$ *vertices.*

Proof. Consider an instance $(\mathcal{M}_{\mathcal{E}}, k)$ of GSCS. Let $\overline{G}$ denote the complement graph of underlying undirected graph of $\mathcal{M}_{\mathcal{E}}$. We first find a vertex cover, X, of $\overline{G}$ of size at most $2q$ using factor 2-approximation algorithm for Vertex Cover [3]. Note that X is a **tvd** of $\mathcal{M}_{\mathcal{E}}$. Therefore, $T = \mathcal{M}_{\mathcal{E}} - X$ is a tournament. Suppose Reduction Rule 7.1 is not applicable, then $|V(T)| \leq k + 1$. If $k \leq 2q$, then using Lemma 7.3, $|V(\mathcal{M}_{\mathcal{E}})| \leq 4q + 1$. Now, suppose that $k > 2q$. Then, Reduction Rule 7.2 or 7.3 is applicable. After exhaustive application of Reduction Rules 7.2 and 7.3, either we return a no-instance of constant size or $k \leq 2q$. As argued above if $k \leq 2q$, we have that $|V(\mathcal{M}_{\mathcal{E}})| \leq 4q + 1$. Note that each of the reduction rules can be applied in the polynomial time, and each of them either declare that the given instance is a no instance or reduces the size of the graph. Therefore, the overall running time is polynomial in the input size. □

Recall that l is the number of missing arcs between the vertices in $\mathcal{M}_{\mathcal{E}}$. Let X be a set of vertices obtained by adding a vertex from each of the missing arcs. Note that X is a **tvd** of $\mathcal{M}_{\mathcal{E}}$, and $|X| \leq l$. Since q is the size of **tvd** of $\mathcal{M}_{\mathcal{E}}$, $q \leq l$, we get the following as a corollary to Theorem 7.5.

Corollary 7.1. GSCS *admits a kernel with* $2l + 1$ *vertices, where* l *is the number of missing arcs in the majority graph.*

8 CONCLUSION

In this paper we studied Gehrlein Stable Committee Selection problem in the realm of parameterized complexity. We put forward a parameterized complexity map of the problem, by way of W-hardness, fixed-parameterized tractability, and kernelization. We showed that the problem is W[1]-hard when parameterized by the size of the committee, yet it admit fixed parameter tractable algorithms and linear kernels with respect to alternate structural parameters which encode the "closeness" of the underlying majority graph of the given election to a tournament. It would be interesting to study parameterized complexity of the committee selection problem under domain restrictions and with respect to other voting rules. Another natural direction is to study the problem with respect to other relevant parameters.

ACKNOWLEDGEMENT

We thank one of the reviewer of an earlier version of the paper for suggesting the current version of the proof of Theorem 4.2, which is simpler than our earlier proof. Pallavi Jain was supported by SERB-NPDF fellowship (PDF/2016/003508) of DST, India. Meirav Zehavi was supported by ISF (1176/18).

REFERENCES

[1] H. Aziz, E. Elkind, P. Faliszewski, M. Lackner, and Piotr Skowron. 2017. The condorcet principle for multiwinner elections: from shortlisting to Proportionality. In *IJCAI*. 84–90.
[2] H. Aziz, E. Elkind, P. Faliszewski, M. Lackner, and P. Skowron. 2017. The condorcet principle for multiwinner elections: from shortlisting to Proportionality. *CoRR* abs/1701.08023 (2017).
[3] R. Bar-Yehuda and S. Even. 1981. A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms* 2, 2 (1981), 198–203.
[4] S. J. Brams, D. M. Kilgour, and M. R. Sanver. 2007. A minimax procedure for electing committees. *Public Choice* 132, 3-4 (2007), 401–420.
[5] J. Chen, I. A. Kanj, and G. Xia. 2010. Improved upper bounds for vertex cover. *Theoretical Computer Science* 411, 40-42 (2010), 3736–3756.
[6] D. Coelho. 2005. *Understanding, evaluating and selecting voting rules through games and axioms*. Universitat Autònoma de Barcelona,.
[7] M. D. Condorcet. 1785. Essai sur l'application de l'analyse, a la probabilite des decisions rendues a la pluralite des voix. (1785).
[8] M. Cygan, F. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. 2015. *Parameterized algorithms*. Springer.
[9] A. Darmann. 2013. How hard is it to tell which is a Condorcet committee? *Mathematical Social Sciences* 66, 3 (2013), 282–292.
[10] M. Dummett. 1984. Voting procedures. (1984).
[11] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. 2017. Properties of multiwinner voting rules. *Social Choice and Welfare* 48, 3 (2017), 599–632.
[12] E. Elkind, J. Lang, and A. Saffidine. 2011. Choosing collectively optimal sets of alternatives based on the condorcet criterion. In *IJCAI*, Vol. 11. 186–191.
[13] E. Elkind, J. Lang, and A. Saffidine. 2015. Condorcet winning sets. *Social Choice and Welfare* 44, 3 (2015), 493–517.
[14] P. Erdős and L. Moser. 1964. On the representation of directed graphs as unions of orderings. *Math. Inst. Hung. Acad. Sci* 9 (1964), 125–132.
[15] P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. 2017. Multiwinner voting: A new challenge for social choice theory. *Trends in Computational Social Choice* 74 (2017).
[16] R. Fellows, D. Hermelin, F. A. Rosamond, and St. Vialette. 2009. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.* 410, 1 (2009), 53–61.
[17] P. C. Fishburn. 1981. An analysis of simple voting systems for electing committees. *SIAM J. Appl. Math.* 41, 3 (1981), 499–502.
[18] Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. 2018. *Kernelization: theory of parameterized preprocessing*. Cambridge University Press.
[19] W. V. Gehrlein. 1985. The Condorcet criterion and committee selection. *Mathematical Social Sciences* 10, 3 (1985), 199–209.
[20] P. Hell and M. Rosenfeld. 1983. The Complexity of Finding Generalized Paths in Tournaments. *Journal of Algorithms* 4, 4 (1983), 303–309.
[21] J. B. Jensen and P. Hell. 1993. Fast algorithms for finding hamiltonian paths and cycles in in-tournament digraphs. *Discrete Applied Mathematics* 41, 1 (1993), 75–79.
[22] E. Kamwa. 2017. On stable rules for selecting committees. *Journal of Mathematical Economics* 70 (2017), 36–44.
[23] B. Kaymak and M. R. Sanver. 2003. Sets of alternatives as Condorcet winners. *Social Choice and Welfare* 20, 3 (2003), 477–494.
[24] J. M. Kleinberg and É. Tardos. 2006. *Algorithm design*. Addison-Wesley.
[25] J.F. Laslier. 2011. *Tournament Solutions and Majority Voting*. Springer Berlin Heidelberg. https://books.google.co.in/books?id=2bJ_MQEACAAJ
[26] S. Salvador and D. Coelho. 2008. How to choose a non-controversial list with k names. *Social Choice and Welfare* 31, 1 (2008), 79–96.