Algorithms for Gerrymandering over Graphs*

Takehiro Ito† Naoyuki Kamiyama‡ Yusuke Kobayashi§ Yoshio Okamoto¶

February 18, 2021

Abstract

We initiate the systematic algorithmic study for gerrymandering over graphs that was recently introduced by Cohen-Zemach, Lewenberg and Rosenschein. Namely, we study a strategic procedure for a political districting designer to draw electoral district boundaries so that a particular target candidate can win in an election. We focus on the existence of such a strategy under the plurality voting rule, and give interesting contrasts which classify easy and hard instances with respect to polynomial-time solvability. For example, we prove that the problem for trees is strongly **NP**-complete (thus unlikely to have a pseudo-polynomial-time algorithm), but has a pseudo-polynomial-time algorithm when the number of candidates is constant. Another example is to prove that the problem for complete graphs is **NP**-complete when the number of electoral districts is two, while is solvable in polynomial time when it is more than two.

Keywords: Gerrymandering; Computational Social Choice; Graph Algorithms

1 Introduction

Control in voting is one of the main topics in computational social choice. For example, Faliszewski and Rothe [12] dedicated one chapter on "Control and Bribery in Voting" for *Handbook of Computational Social Choice*, and gave an overview of the topic. One of the earliest papers was written by Bartholdi, Tovey, and Trick [17] who studied the manipulability of elections from the viewpoint of computational complexity. Among others, they studied the manipulation of the election result by partitioning the set of voters. They called the problem "*Control by Partition of Voters*," but in fact, this is quite similar to the problem that is usually called *gerrymandering* in the political geography literature.

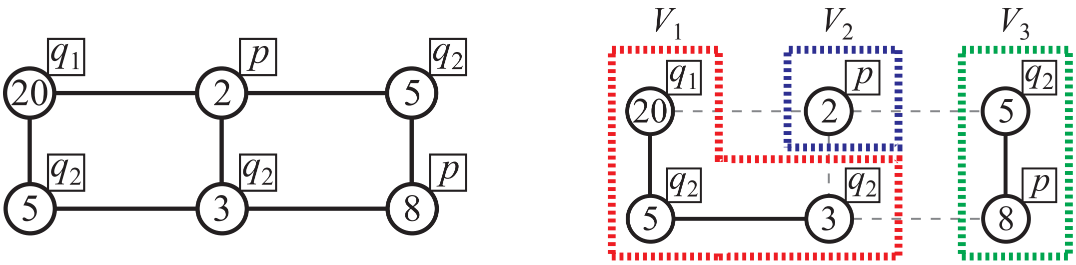
We study the gerrymandering model that is proposed by Cohen-Zemach, Lewenberg and Rosenschein [7]. For brevity, we describe their model only for the *plurality voting rule*, which we adopt in this paper. Namely, we consider a hierarchical voting process as follows. The set of voters is partitioned into several groups, and each of the groups holds an independent election. From each group, one candidate is elected as a nominee. Then, among the elected nominees, a final voting is held to determine the winner. In the *plurality voting rule*, a candidate who gets the plurality votes is a nominee in the first stage, and a nominee who won in the most groups is a final winner.

Gerrymandering is a word that means a strategic procedure for a political districting designer to draw electoral district boundaries so that the outcome of the election can be under control. Typically, such control implies the win of a particular candidate in the election. Gerrymandering is considered a bad practice, and one of the main motivations of research in political (re)districting is to avoid gerrymandering.

To model geographic constraints, Cohen-Zemach et al. [7] used a network structure, i.e., an undirected graph. Cohen-Zemach et al. [7] called the framework the *gerrymandering over graphs*. In gerrymandering over graphs, we are given an undirected graph $G = (V, E)$, a natural number k, a set $\mathcal{C}$ of candidates, a target candidate $p \in \mathcal{C}$, the weight $w(v)$ of each vertex $v \in V$, and a candidate $c(v)$ preferred by

*A preliminary version has appeared in Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '19), pp. 1413–1421.

†Tohoku University, Sendai, Japan. takehiro@ecei.tohoku.ac.jp.
‡Kyushu University, Fukuoka, Japan and JST, PRESTO, Kawaguchi, Japan. kamiyama@imi.kyushu-u.ac.jp.
§Kyoto University, Kyoto, Japan. yusuke@kurims.kyoto-u.ac.jp.
¶University of Electro-Communications, Chofu, Japan. okamotoy@uec.ac.jp.



(a) (b)

Figure 1: (a) Input graph $G = (V, E)$, $\mathcal{C} = \{p, q_1, q_2\}$ and $k = 3$, where the weight $w(v)$ of each vertex v is written inside the vertex (circle) and the candidate $c(v) \in \mathcal{C}$ preferred by v is written inside the square attached to v. (b) A desired partition of V into $k = 3$ parts V_1, V_2, V_3. In the first stage of the voting process, q_1 wins in V_1 and the target candidate p wins in V_2 and V_3. Thus, p is elected in the second stage as the final winner.

each vertex $v \in V$. See also Figure 1. We want to decide if there exists a partition of V into exactly k non-empty parts $V_1, V_2, \ldots, V_k$ such that (1) each part in the partition induces a connected subgraph of G and (2) the number of parts in which p wins is larger than the number of parts in which any other candidate wins. Section 2 will give a more formal description.

The contributions of their paper [7] were two-fold. First, they proved that it is **NP**-complete to decide if there is a partition of a given graph such that each part contains at least two vertices and the target candidate p wins in at least b parts, for a given positive integer b. Second, they conducted simulation studies on random graphs and real-world networks for their original problem setting.

1.1 Our results

In this paper, we pursue theoretical studies of gerrymandering over graphs from the algorithmic point of view, and give a more systematic treatment to the problem. More specifically, we aim at classifying easy and hard instances of gerrymandering over graphs with respect to polynomial-time solvability. The results are summarized as follows.

On the negative side, we prove that the problem is **NP**-complete even for very restricted cases. First, we prove the hardness even when $k = 2$, $|\mathcal{C}| = 2$, and G is complete. The same hardness also applies when G is a planar graph of pathwidth two ($K_{2,n}$). Second, we prove the hardness when all vertex weights are identical and $|\mathcal{C}| = 4$. Third, we prove that the problem is strongly **NP**-complete when G is a tree of diameter four (thus, cannot be solved in pseudo-polynomial time unless $\mathbf{P} = \mathbf{NP}$).

On the positive side, we provide polynomial-time algorithms for the following special cases of trees. First, we solve the problem for stars (i.e., trees of diameter two) in polynomial time. Second, we give a polynomial-time algorithm for paths when $|\mathcal{C}|$ is constant. Third, we give a pseudo-polynomial-time algorithm for trees when $|\mathcal{C}|$ is constant; this gives an interesting contrast to the strong **NP**-completeness for trees when $|\mathcal{C}|$ is a part of the input. We note that it is easy to see that the problem can be solved in polynomial time for trees when k is constant (nevertheless, we give a proof for completeness).

As another interesting contrast, we give a polynomial-time algorithm for complete graphs when $k \geq 3$; recall that the problem is **NP**-complete when $k = 2$. We also give a pseudo-polynomial-time algorithm when $k = 2$.

We note that the following two cases are unsettled: a polynomial-time algorithm for paths (when $|\mathcal{C}|$ is not constant) and one for trees (when $|\mathcal{C}|$ is constant). They form main open problems from this paper.

1.2 Past Work

As mentioned before, control in voting is one of the major topics in computational social choice theory. After the paper by Bartholdi, Tovey, and Trick [17], numerous authors studied several variants, e.g., [16, 15, 11, 19, 3, 9, 10, 1, 2, 22, 4].

To cope with gerrymandering, several authors have studied the political (re)districting problem. In the political districting problem, we are given a geographic region with population, and want to partition the region into several parts as to satisfy given constraints such as the shape of each part, small variance of the populations among parts, etc. In the operations research literature, heuristic algorithms have been developed, e.g., [20, 5, 23, 6]. To the best of the authors' knowledge, there seems no algorithm with a theoretical guarantee for the quality of the output.

As theoretical studies for gerrymandering, we are aware of four papers in which **NP**-hardness is proved. Puppe and Tasnádi [21] treated geographic constraints by combinatorics (i.e., certain sets of voters cannot form parts in the partition). Fleiner, Nagy and Tasnádi [13] and Lewenberg, Lev and Rosenschein [18] treated geographic constraints by geometry, where each group needs to be induced by a simply connected region in the plane in [13], and each group is determined by a closest ballot box in [18]. Cohen-Zemach, Lewenberg and Rosenschein [7] treated geographic constraints by networks, and each group needs to be induced by a connected subgraph. We adopt the model by Cohen-Zemach et al. in this paper.

1.3 Organization of the Paper

We start with the formal problem description in Section 2. The **NP**-completeness is discussed in Section 3. Algorithms for trees are given in Section 4. We provide algorithms for complete graphs in Section 5, and conclude the paper in Section 6.

2 Problem Description

Let $G = (V, E)$ be an undirected graph. For a positive integer k, a partition of V into non-empty k subsets $V_1, V_2, \ldots, V_k$ is called a *connected partition* of G if the induced subgraph $G[V_i]$ is connected for every $i \in \{1, 2, \ldots, k\}$. We sometimes call each connected component $G[V_i]$ a *constituency* in the connected partition of G.

Let $\mathcal{C}$ be a finite set called the set of *candidates*. One element p of $\mathcal{C}$ is designated as the *target* candidate. We often denote $\mathcal{C} = \{p, q_1, q_2, \ldots, q_\ell\}$. Each vertex $v \in V$ has an associated positive integer weight $w(v)$, and an associated candidate $c(v) \in C$ that the vertex v prefers. Since each vertex v prefers only one candidate $c(v)$, we assume without loss of generality that $|\mathcal{C}| \leq |V|$. For a vertex subset $U \subseteq V$ of G, the set of all candidates that receive the largest total weight in U is denoted by $\mathsf{top}(U)$, that is,

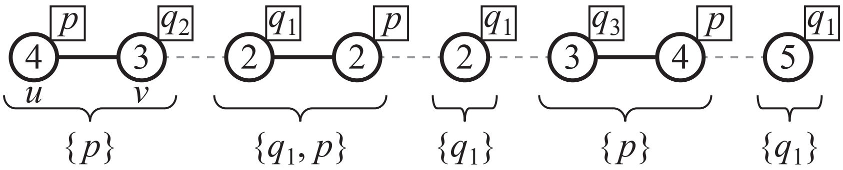
$$\mathsf{top}(U) \coloneqq \arg\max_{q \in \mathcal{C}} \left\{ \sum_{v \in U\,:\, c(v) = q} w(v) \right\}.$$

See also Figure 2. An element of $\mathsf{top}(U)$ is often referred to as a *top candidate* in U (or in $G[U]$). We sometimes say that a candidate $q \in \mathcal{C}$ *wins* in a constituency $G[U]$ if $q \in \mathsf{top}(U)$; in particular, $q \in \mathcal{C}$ *wins alone* in $G[U]$ if $\mathsf{top}(U) = \{q\}$.

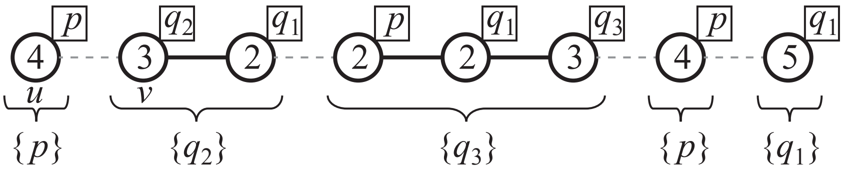
The gerrymandering problem over a graph can be formulated as follows. We are given an undirected graph $G = (V, E)$, the set $\mathcal{C}$ of candidates, the target candidate $p \in \mathcal{C}$, and a positive integer k. For each vertex $v \in V$, we are also given an associated positive integer weight $w(v)$ and an associated candidate $c(v)$. Then, we want to decide if there exists a connected partition of G into k parts $V_1, V_2, \ldots, V_k$ such that p is the unique top candidate in the most constituencies of the partition; namely

$$|\{i \in \{1, 2, \ldots, k\} : \{p\} = \mathsf{top}(V_i)\}| > |\{i \in \{1, 2, \ldots, k\} : q \in \mathsf{top}(V_i)\}| \quad \forall\, q \in \mathcal{C} \setminus \{p\}.$$

The left-hand side represents the number of constituencies in which p wins alone, and the right-hand side represents the number of constituencies in which q is one of the top candidates. Therefore, the condition means that in the connected partition $V_1, V_2, \ldots, V_k$ of G, the target candidate p can win in the most constituencies no matter which tie-breaking rule is adopted among the top candidates. Such a connected partition of G is often referred to as a *feasible solution* in this paper. See also Figure 2. Note that if $|V| < k$ or G has more than k connected components, then we can immediately conclude that there is no feasible solution.



(a)



(b)

Figure 2: (a) A connected partition of a path G (which is not a feasible solution), and (b) a feasible solution, where $k = 5$, p is the target candidate, and $\mathsf{top}(V_i)$ is written below each constituency V_i.

Algorithmic Complexity An algorithm is said to be pseudo-polynomial-time if its running time is bounded by a polynomial in the numerical values of the input. A problem is said to be strongly **NP**-complete if it remains **NP**-complete even when the numerical values of the input are bounded by a polynomial in the encoding length of the input. Thus, a strongly **NP**-complete problem does not admit a pseudo-polynomial-time algorithm unless **P** = **NP**.

3 Hardness of Gerrymandering

In this section, we prove that the gerrymandering problem is computationally intractable even for very restricted cases.

3.1 Hardness via Partition

We first consider the case where both k and $|\mathcal{C}|$ are fixed to two.

Theorem 1. *The gerrymandering problem is* **NP***-complete when* $k = 2$*,* $|\mathcal{C}| = 2$*, and* G *is either a complete bipartite graph* $K_{2,n}$ *or a complete graph.*

Proof. We give a polynomial-time reduction from Partition: an instance is given by a list of n positive integers $a_1, a_2, \ldots, a_n$, and the problem asks to decide if there exists a set $S \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in S} a_i = \sum_{i \notin S} a_i$. It is known [14] that Partition is **NP**-complete. We now construct an instance of the gerrymandering problem. Let $G = (U, V; E)$ be a complete bipartite graph with $U \coloneqq \{u_1, u_2\}$ and $V \coloneqq \{v_1, v_2, \ldots, v_n\}$. For each $v \in U \cup V$, we define

$$w(v) \coloneqq \begin{cases} \varepsilon + \dfrac{1}{2}\displaystyle\sum_{i=1}^{n} a_i & \text{if } v \in U; \\ a_i & \text{if } v = v_i \text{ for } i \in \{1, 2, \ldots, n\}, \end{cases}$$

where $\varepsilon = \frac{1}{3}$. We note that we can make each $w(v)$ an integer by scaling the weight function, but we use the fractional weight function as above to simplify the description. Let $\mathcal{C} \coloneqq \{p, q\}$, where p is the target candidate, and define $c(v) \coloneqq p$ if $v \in U$, and $c(v) \coloneqq q$ if $v \in V$. Let $k \coloneqq 2$.

For the **NP**-completeness on complete graphs, we join every pair of vertices in the bipartite graph $G = (U, V; E)$ above.

Since the membership in **NP** is easy, to complete the proof of Theorem 1, it suffices to prove the following claim.

Claim 1. *The original instance of Partition has a desired set $S \subseteq \{1, 2, \ldots, n\}$ if and only if the corresponding instance of the gerrymandering problem has a feasible solution.*

We note that all arguments below hold for both complete bipartite graphs $K_{2,n}$ and complete graphs.

We first prove the necessity. If $S \subseteq \{1, 2, \ldots, n\}$ satisfies that $\sum_{i \in S} a_i = \sum_{i \notin S} a_i$, then we define $V_1 \coloneqq \{u_1\} \cup \{v_i : i \in S\}$ and $V_2 \coloneqq \{u_2\} \cup \{v_i : i \notin S\}$. Then, (V_1, V_2) is a partition of $U \cup V$ such that $G[V_j]$ is connected and $\mathsf{top}(V_j) = \{p\}$ for $j \in \{1, 2\}$. Therefore, (V_1, V_2) is a feasible solution to the gerrymandering problem.

To show the sufficiency, suppose that (V_1, V_2) is a feasible solution to the gerrymandering problem. Since $k = 2$, it holds that $|\{j \in \{1, 2\} : \{p\} = \mathsf{top}(V_j)\}| = 2$, that is, $\mathsf{top}(V_1) = \mathsf{top}(V_2) = \{p\}$. Recall that only two vertices u_1 and u_2 prefer the target candidate p. Since $\mathsf{top}(V_1) = \mathsf{top}(V_2) = \{p\}$, we have $V_j \cap \{u_1, u_2\} \neq \emptyset$ for each $j \in \{1, 2\}$; we may thus assume that $u_1 \in V_1$ and $u_2 \in V_2$. Let $S \coloneqq \{i \in \{1, 2, \ldots, n\} : v_i \in V_1\}$. Then, $\mathsf{top}(V_1) = \mathsf{top}(V_2) = \{p\}$ implies that

$$\sum_{i \in S} a_i = \sum_{v \in V \cap V_1} w(v) < w(u_1) = \varepsilon + \frac{1}{2}\sum_{i=1}^{n} a_i,$$
$$\sum_{i \notin S} a_i = \sum_{v \in V \cap V_2} w(v) < w(u_2) = \varepsilon + \frac{1}{2}\sum_{i=1}^{n} a_i.$$

By these inequalities, we have that $\sum_{i \in S} a_i = \frac{1}{2}\sum_{i=1}^{n} a_i = \sum_{i \notin S} a_i$, which shows that S is a desired set to Partition. ☐

We note that a complete bipartite graph $K_{2,n}$ is of pathwidth two. Thus, the gerrymandering problem remains **NP**-complete even for bounded pathwidth graphs and $k = |\mathcal{C}| = 2$.

We also note that the above **NP**-completeness proof works also for the case with $k = 2$ and $|\mathcal{C}| \geq 3$. To see this, let $\mathcal{C} \coloneqq \{p, q, q_1, \ldots, q_\ell\}$, construct a complete graph or a complete bipartite graph as in the proof of Theorem 1, and add a new vertex u_i with $c(u_i) = q_i$ and $w(u_i) = \frac{1}{3}$ together with appropriate incident edges for each $i \in \{1, 2, \ldots, \ell\}$. Since $u_1, \ldots, u_\ell$ do not affect the arguments in the proof of Theorem 1, we obtain the following corollary.

Corollary 1. *The gerrymandering problem is* **NP***-complete when $k = 2$, $|\mathcal{C}| \geq 3$, and G is either a complete bipartite graph $K_{2,n}$ or a complete graph.*

In contrast to the **NP**-completeness on complete graphs for $k = 2$, we will prove in Section 5 that the problem is solvable in polynomial time if G is a complete graph and $k \geq 3$; note that $|\mathcal{C}|$ is not necessarily fixed.

When there is no restriction on G, the **NP**-completeness proof can be extended to the case with $k \geq 2$. To see this, construct a complete graph or a complete bipartite graph as in the proof of Theorem 1 (or Corollary 1) and add a set R of $k - 2$ isolated vertices such that $|\{v \in R : c(v) = p\}| = \lfloor \frac{k-2}{2} \rfloor$ and $|\{v \in R : c(v) = q\}| = \lceil \frac{k-2}{2} \rceil$. Let $G = (V, E)$ be the obtained graph. Then, the obtained instance is equivalent to finding a connected partition (V_1, V_2) of $G[V \setminus R]$ such that $\mathsf{top}(V_1) = \mathsf{top}(V_2) = \{p\}$, which is exactly the same as Theorem 1 (or Corollary 1). This shows the following corollary.

Corollary 2. *The gerrymandering problem is* **NP***-complete for any fixed $k \geq 2$ and any fixed $|\mathcal{C}| \geq 2$.*

3.2 Hardness for Unit Weight Case via 3-Partition

We then consider the case where every vertex has a unit weight.

Theorem 2. *The gerrymandering problem is* **NP***-complete even if $w(v) = 1$ for every $v \in V$ and $|\mathcal{C}| = 4$.*

Proof. We give a polynomial-time reduction from 3-Partition: given a list of $3n$ positive integers $a_1, a_2, \ldots, a_{3n}$ as an instance, the problem asks to decide if there exists a partition $S_1, S_2, \ldots, S_n$ of $\{1, 2, \ldots, 3n\}$ such that $\sum_{i \in S_j} a_i = \frac{1}{n}\sum_{i=1}^{3n} a_i$ for every $j \in \{1, 2, \ldots, n\}$. It is known that 3-Partition remains **NP**-complete even when each integer a_i is bounded by some polynomial in n (see, e.g., [14]). We may assume that $t \coloneqq \frac{1}{n}\sum_{i=1}^{3n} a_i$ is an integer, since otherwise we can immediately conclude that there exists no solution, because $\sum_{i \in S_j} a_i$ is an integer.

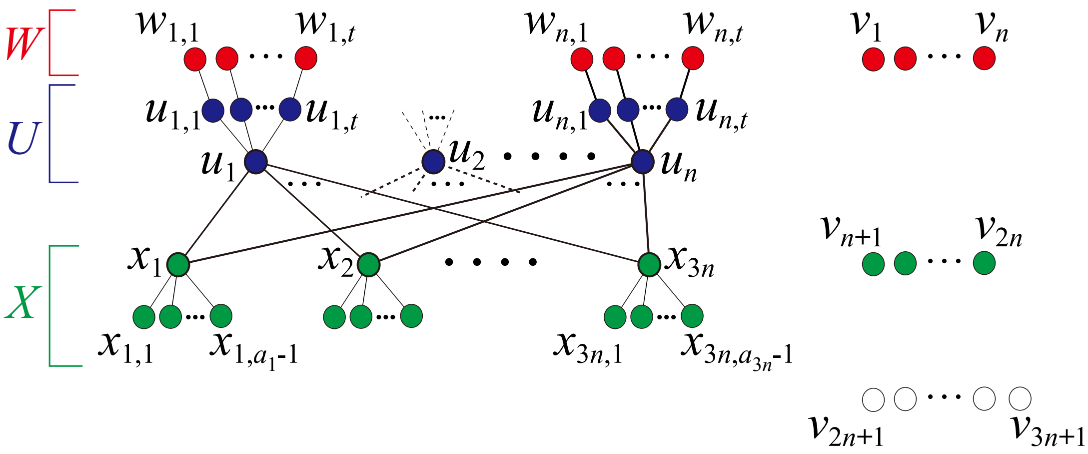


Figure 3: Construction for Theorem 2.

We construct an instance of the gerrymandering problem. As Figure 3 illustrates, consider a graph $G = (V, E)$ defined as follows:

$$
\begin{aligned}
U &\coloneqq \{u_1, u_2, \ldots, u_n\} \cup \{u_{i,h} : i \in \{1, 2, \ldots, n\},\ h \in \{1, 2, \ldots, t\}\},\\
W &\coloneqq \{w_{i,h} : i \in \{1, 2, \ldots, n\},\ h \in \{1, 2, \ldots, t\}\} \cup \{v_1, v_2, \ldots, v_n\},\\
X &\coloneqq \{x_1, x_2, \ldots, x_{3n}\} \cup \{x_{i,h} : i \in \{1, 2, \ldots, 3n\},\ h \in \{1, 2, \ldots, a_i - 1\}\} \cup \{v_{n+1}, v_{n+2}, \ldots, v_{2n}\},\\
V &\coloneqq U \cup W \cup X \cup \{v_{2n+1}, v_{2n+2}, \ldots, v_{3n+1}\},\\
E &\coloneqq \{(u_i, u_{i,h}), (u_{i,h}, w_{i,h}) : i \in \{1, 2, \ldots, n\},\ h \in \{1, 2, \ldots, t\}\}\\
&\quad \cup \{(x_i, x_{i,h}) : i \in \{1, 2, \ldots, 3n\},\ h \in \{1, 2, \ldots, a_i - 1\}\}\\
&\quad \cup \{(x_i, u_j) : i \in \{1, 2, \ldots, 3n\},\ j \in \{1, 2, \ldots, n\}\}.
\end{aligned}
$$

Let $\mathcal{C} \coloneqq \{p, q_1, q_2, q_3\}$, where p is the target candidate. For each $v \in V$, we define $c(v)$ as

$$
c(v) \coloneqq \begin{cases} p & \text{if } v = v_i \text{ for some } i \in \{2n+1, 2n+2, \ldots, 3n+1\},\\ q_1 & \text{if } v \in U,\\ q_2 & \text{if } v \in W,\\ q_3 & \text{if } v \in X. \end{cases}
$$

Let $k = 4n + 1$.

Since the membership in **NP** is easy, to complete the proof of Theorem 2, it suffices to prove the following claim.

Claim 2. *The original instance of 3-Partition has a desired partition $S_1, S_2, \ldots, S_n$ if and only if the corresponding instance of the gerrymandering problem has a feasible solution.*

We first show the necessity. Assume that the original instance of 3-Partition has a desired partition $S_1, S_2, \ldots, S_n$ of $\{1, 2, \ldots, 3n\}$. We define a partition $V_1, V_2, \ldots, V_{4n+1}$ of V, as follows: Define

$$
V_j \coloneqq \{u_j\} \cup \{u_{j,h}, w_{j,h} : h \in \{1, 2, \ldots, t\}\} \cup \{x_i : i \in S_j\} \cup \{x_{i,h} : i \in S_j,\ h \in \{1, 2, \ldots, a_i - 1\}\}
$$

for each $j \in \{1, 2, \ldots, n\}$, and define $V_j \coloneqq \{v_{j-n}\}$ for each $j \in \{n+1, n+2, \ldots, 4n+1\}$. Then, each $G[V_j]$ is connected, and hence $(V_1, V_2, \ldots, V_{4n+1})$ forms a connected partition of G. Furthermore, $\mathsf{top}(V_j) = \{q_1\}$ holds for all $j \in \{1, 2, \ldots, n\}$, because we have

- $|\{v \in V_j : c(v) = q_1\}| = t + 1$,
- $|\{v \in V_j : c(v) = q_i\}| = t$ for each $i \in \{2, 3\}$, and

- $|\{v \in V_j : c(v) = p\}| = 0$.

Similarly, we can see that

- $\mathsf{top}(V_j) = \{q_2\}$ for each $j \in \{n+1, n+2, \ldots, 2n\}$,
- $\mathsf{top}(V_j) = \{q_3\}$ for each $j \in \{2n+1, 2n+2, \ldots, 3n\}$, and
- $\mathsf{top}(V_j) = \{p\}$ for each $j \in \{3n+1, 3n+2, \ldots, 4n+1\}$,

because $V_j = \{v_{j-n}\}$. Therefore, we obtain

$$|\{j \in \{1,2,\ldots,k\} : \{p\} = \mathsf{top}(V_j)\}| > |\{j \in \{1,2,\ldots,k\} : q \in \mathsf{top}(V_j)\}| \quad \forall\, q \in \mathcal{C} \setminus \{p\},$$

which shows the necessity.

We next show the sufficiency. Assume that there exists a connected partition $\mathcal{V} = (V_1, V_2, \ldots, V_k)$ of G that is a feasible solution to the gerrymandering problem. Since $\{v_j\}$ forms a part of $\mathcal{V}$, say V_{n+j}, for $j \in \{1, 2, \ldots, 3n+1\}$, $V \setminus \{v_1, v_2, \ldots, v_{3n+1}\}$ is partitioned into n sets $V_1, V_2, \ldots, V_n$ such that $G[V_j]$ is connected for each $j \in \{1, 2, \ldots, n\}$. By the construction of G, we obtain

- $|\{j \in \{1,2,\ldots,k\} : \{p\} = \mathsf{top}(V_j)\}| = |\{j \in \{n+1,n+2,\ldots,k\} : \{p\} = \mathsf{top}(V_j)\}| = n+1$,
- $|\{j \in \{1,2,\ldots,k\} : q_2 \in \mathsf{top}(V_j)\}| \geq |\{j \in \{n+1,n+2,\ldots,k\} : q_2 \in \mathsf{top}(V_j)\}| = n$,
- $|\{j \in \{1,2,\ldots,k\} : q_3 \in \mathsf{top}(V_j)\}| \geq |\{j \in \{n+1,n+2,\ldots,k\} : q_3 \in \mathsf{top}(V_j)\}| = n$.

Since

$$|\{j \in \{1,2,\ldots,k\} : \{p\} = \mathsf{top}(V_j)\}| > |\{j \in \{1,2,\ldots,k\} : q \in \mathsf{top}(V_j)\}| \quad \forall\, q \in \mathcal{C} \setminus \{p\}$$

by the feasibility of $\mathcal{V}$, we have that $q_2, q_3 \notin \mathsf{top}(V_j)$ for each $j \in \{1, 2, \ldots, n\}$, that is, $\mathsf{top}(V_j) = \{q_1\}$ for $j \in \{1, 2, \ldots, n\}$. Since $\mathsf{top}(V_j) = \{q_1\}$, V_j contains at least one vertex in U for each $j \in \{1, 2, \ldots, n\}$. Due to the connectedness of $G[V_j]$, without loss of generality, we may assume that $u_j \in V_j$. This also implies that $\{u_{j,h}, w_{j,h} : h \in \{1, 2, \ldots, t\}\} \subseteq V_j$.

For $j \in \{1, 2, \ldots, n\}$, define $S_j \coloneqq \{i \in \{1, 2, \ldots, 3n\} : x_i \in V_j\}$. Then, since $x_i \in V_j$ implies $\{x_i\} \cup \{x_{i,h} : h \in \{1, 2, \ldots, a_i - 1\}\} \subseteq V_j$, we have

$$V_j = \{u_i\} \cup \{u_{i,h}, w_{i,h} : h \in \{1,2,\ldots,t\}\} \cup \{x_i : i \in S_j\} \cup \{x_{i,h} : i \in S_j,\ h \in \{1,2,\ldots,a_i - 1\}\}.$$

Since $|\{v \in V_j : c(v) = q_1\}| = t + 1$, $|\{v \in V_j : c(v) = q_3\}| = \sum_{i \in S_j} a_i$, and $\mathsf{top}(V_j) = \{q_1\}$, it holds that $\sum_{i \in S_j} a_i < t + 1$, which implies that $\sum_{i \in S_j} a_i \leq t$ by the integrality of a_i and t. Therefore,

$$\sum_{i=1}^{3n} a_i = \sum_{j=1}^{n} \sum_{i \in S_j} a_i \leq n \cdot t = n \cdot \frac{1}{n} \cdot \sum_{i=1}^{3n} a_i = \sum_{i=1}^{3n} a_i.$$

Hence, we obtain $\sum_{i \in S_j} a_i = \frac{1}{n} \sum_{i=1}^{3n} a_i$ for $j = 1, 2, \ldots, n$, which shows that the original instance of 3-Partition has a desired partition. This completes the proof of the sufficiency. ☐

We note that the graph in the reduction can be made connected, because the same argument works even if we add an edge between v_{3n+1} and v for every $v \in V \setminus \{v_{3n+1}\}$.

3.3 Hardness for Trees via Satisfiability

We finally consider the case for trees.

Theorem 3. *The gerrymandering problem is strongly* **NP***-complete even for trees of diameter four.*

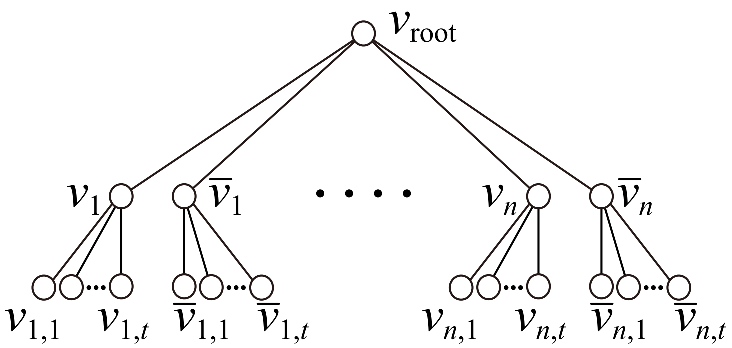


Figure 4: Construction for Theorem 3.

Proof. We give a polynomial-time reduction from 3-SAT. Consider an instance of 3-SAT with n (≥ 2) variables $x_1, x_2, \ldots, x_n$ and m clauses $C_1, C_2, \ldots, C_m$, in which each clause contains exactly three distinct literals. It is well-known that this problem is **NP**-complete (see, e.g., [14]). Furthermore, we may assume that n is odd, since we can add a new variable that appears in none of the clauses.

We construct an instance of the gerrymandering problem. Set $t \coloneqq n-1+\frac{m(n-1)}{2}$ and $k \coloneqq n(t+1)+1$. As Figure 4 illustrates, consider a tree $G = (V, E)$ defined as follows:

$$V \coloneqq \{v_{\text{root}}\} \cup \{v_i, \bar{v}_i : i \in \{1, 2, \ldots, n\}\} \cup \{v_{i,j}, \bar{v}_{i,j} : i \in \{1, 2, \ldots, n\},\ j \in \{1, 2, \ldots, t\}\},$$
$$E \coloneqq \{(v_{\text{root}}, v_i), (v_{\text{root}}, \bar{v}_i) : i \in \{1, 2, \ldots, n\}\} \cup \{(v_i, v_{i,j}), (\bar{v}_i, \bar{v}_{i,j}) : i \in \{1, 2, \ldots, n\},\ j \in \{1, 2, \ldots, t\}\}.$$

We regard G as a rooted tree with the root v_{root}. Let M be a sufficiently large integer (e.g., $M = |V| + 1$), and define the weight of each vertex as

$$w(v) \coloneqq \begin{cases} M^2 & \text{if } v = v_{\text{root}}; \\ 1 & \text{if } v = v_i \text{ or } v = \bar{v}_i \text{ for some } i \in \{1, 2, \ldots, n\}; \\ M & \text{otherwise.} \end{cases}$$

We note that the weight of each vertex is bounded by a polynomial in $|V|$. Define the set $\mathcal{C}$ of candidates as

$$\mathcal{C} \coloneqq \{p, q_1, \ldots, q_n, r_1, \ldots, r_m\} \cup \{s_{\text{root}}\} \cup \{s_{i,j} : i \in \{1, 2, \ldots, n\},\ j \in \{1, 2, \ldots, t\}\}.$$

Here, p is the target candidate, while q_i and r_j correspond to the variable x_i and the clause C_j, respectively. The candidates s_{root} and $s_{i,j}$ will act as dummy candidates. Define $c(v_{i,j})$ for each leaf $v_{i,j}$ of G as follows.

- For each $i \in \{1, 2, \ldots, n\}$, pick up $n-1$ children of v_i and associate them with q_i, that is,

  $$|\{v \in V : v \text{ is a child of } v_i,\ c(v) = q_i\}| = n - 1.$$

  Similarly, pick up $n-1$ children of $\bar{v}_i$ and associate them with q_i.

- If C_j contains x_i for $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, m\}$, then pick up $\frac{n-1}{2}$ children of $\bar{v}_i$ and associate them with r_j, that is, $|\{v \in V : v \text{ is a child of } \bar{v}_i,\ c(v) = r_j\}| = \frac{n-1}{2}$.

- If C_j contains $\bar{x}_i$ for $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, m\}$, then pick up $\frac{n-1}{2}$ children of v_i and associate them with r_j, that is, $|\{v \in V : v \text{ is a child of } v_i,\ c(v) = r_j\}| = \frac{n-1}{2}$.

- If $v_{i,j}$ is associated with none of $\{q_1, \ldots, q_n, r_1, \ldots, r_m\}$ in the above procedures, then set $c(v_{i,j}) \coloneqq s_{i,j}$.

Define $c(v_i) \coloneqq p$, $c(\bar{v}_i) \coloneqq p$ for each $i \in \{1, 2, \ldots, n\}$ and $c(v_{\text{root}}) \coloneqq s_{\text{root}}$.

To complete the proof of Theorem 3, we prove the following claim.

Claim 3. *The original instance of 3-SAT has a satisfying truth assignment if and only if the corresponding instance of the gerrymandering problem has a feasible solution.*

We first show the necessity. Assume that the original instance of 3-SAT has a satisfying truth assignment. We construct a partition of V as follows: for $i \in \{1, 2, \dots n\}$, remove all the edges incident to v_i if **True** is assigned to x_i, and remove all the edges incident to $\bar{v}_i$ otherwise. Since the degree of each vertex v_i (or $\bar{v}_i$) is $t + 1$, this operation divides the graph G into $n(t + 1) + 1 = k$ connected components. Let $V_1, V_2, \dots, V_k$ be the vertex sets of these connected components, and consider the partition $\mathcal{V} = (V_1, V_2, \dots, V_k)$. Without loss of generality, we may assume that $v_{\text{root}} \in V_1$ and each of $V_2, V_3, \dots V_k$ consists of a single vertex. Then, we can see the following.

- Since $w(v_{\text{root}})$ is sufficiently large, we have $\mathsf{top}(V_1) = \{s_{\text{root}}\}$.
- Since exactly one of $\{v_i\}$ and $\{\bar{v}_i\}$ is a part of $\mathcal{V}$ for each $i \in \{1, 2, \dots n\}$, we have $|\{h \in \{1, 2, \dots, k\} : \{p\} = \mathsf{top}(V_h)\}| = n$.
- For each $i \in \{1, 2, \dots n\}$, if $\{v_i\}$ or $\{\bar{v}_i\}$ is a part of $\mathcal{V}$, then its each child also forms a part of $\mathcal{V}$. Since exactly one of $\{v_i\}$ and $\{\bar{v}_i\}$ forms a part of $\mathcal{V}$, we have $|\{h \in \{1, 2, \dots, k\} : q_i \in \mathsf{top}(V_h)\}| = n - 1$.
- For each $j \in \{1, 2, \dots m\}$, at least one literal in C_j is assigned **True**. If a literal x_i (resp. $\bar{x}_i$) in C_j is assigned **True**, then all the children of $\bar{v}_i$ (resp. v_i) are contained in V_1. Since $|\{v \in V : v \text{ is a child of } \bar{v}_i \text{ (resp. } v_i), c(v) = r_j\}| = \frac{n-1}{2}$ and $|\{v \in V : c(v) = r_j\}| = \frac{3(n-1)}{2}$, we have $|\{h \in \{1, 2, \dots, k\} : r_j \in \mathsf{top}(V_h)\}| \leq \frac{3(n-1)}{2} - \frac{n-1}{2} = n - 1$.
- For each $i \in \{1, 2, \dots n\}$ and $j \in \{1, 2, \dots m\}$, it is obvious that $|\{h \in \{1, 2, \dots, k\} : s_{i,j} \in \mathsf{top}(V_h)\}| \leq 1$.

Therefore, we obtain

$$|\{h \in \{1, 2, \dots, k\} : \{p\} = \mathsf{top}(V_h)\}| > |\{h \in \{1, 2, \dots, k\} : q \in \mathsf{top}(V_h)\}| \quad \forall\, q \in \mathcal{C} \setminus \{p\},$$

which shows the necessity.

We next show the sufficiency. Assume that there exists a partition $\mathcal{V} = (V_1, V_2, \dots, V_k)$ that is a feasible solution to the gerrymandering problem. Since $c(v) = p$ implies $w(v) = 1$ for any $v \in V$, we can see that if $\{p\} = \mathsf{top}(V_h)$ for $h \in \{1, 2, \dots, k\}$, then either $V_h = \{v_i\}$ or $V_h = \{\bar{v}_i\}$ for some $i \in \{1, 2, \dots, n\}$. Thus, since $\{h \in \{1, 2, \dots, k\} : \{p\} = \mathsf{top}(V_h)\} \neq \emptyset$, there exists $i \in \{1, 2, \dots, n\}$ such that $\{v_i\}$ or $\{\bar{v}_i\}$ is a part of $\mathcal{V}$. Since each child of v_i or $\bar{v}_i$ also forms a part of $\mathcal{V}$, we have $|\{h \in \{1, 2, \dots, k\} : q_i \in \mathsf{top}(V_h)\}| \geq n - 1$, and hence $|\{h \in \{1, 2, \dots, k\} : \{p\} = \mathsf{top}(V_h)\}| \geq n$. This means that $|X| \geq n$, where X is defined as $X \coloneqq \{v \in \{v_1, \bar{v}_1, \dots, v_n, \bar{v}_n\} : \{v\} \text{ is a part of } \mathcal{V}\}$.

In order to make a vertex $v \in X$ isolated, we have to remove all the edges incident to v. Since the degree of $v \in X$ is $t + 1$, the graph G is divided into $|X|(t + 1) + 1$ connected components by removing all the edges incident to a vertex in X. Since $|X| \geq n$ and $k = n(t + 1) + 1$, it holds that $|X| = n$. For each $i \in \{1, 2, \dots, n\}$, if v_i and $\bar{v}_i$ are both in X, then $|\{h \in \{1, 2, \dots, k\} : q_i \in \mathsf{top}(V_h)\}| \geq 2(n - 1) \geq n$, which is a contradiction. Therefore, $|X \cap \{v_i, \bar{v}_i\}| = 1$ for each $i \in \{1, 2, \dots, n\}$. Using this fact, we define an assignment to each variable as follows: we assign **True** to x_i if $v_i \in X$, and assign **False** to x_i if $\bar{v}_i \in X$.

For $j \in \{1, 2, \dots, m\}$, since $|\{v \in V : c(v) = r_j\}| = \frac{3(n-1)}{2} \geq n$, there exists a vertex $v \in V$ with $c(v) = r_j$ that does not form a part of $\mathcal{V}$. That is, we have either x_i is in C_j and $\bar{v}_i \notin X$, or $\bar{x}_i$ is in C_j and $v_i \notin X$ for some $i \in \{1, 2, \dots, n\}$. This shows that C_j contains a literal that is assigned **True** by the definition of the assignment. Therefore, the original instance of 3-SAT has a satisfying truth assignment, which shows the sufficiency. ◻

4 Algorithms for Trees

In contrast to Theorem 3, we show some tractable cases for trees in this section. We first note the following observation.

Theorem 4. *The gerrymandering problem is solvable in polynomial time for trees when k is a fixed constant.*

Proof. Since a given graph $G = (V, E)$ is a tree, we need to delete exactly $k-1$ edges to obtain a partition $V_1, V_2, \ldots, V_k$ of V such that $G[V_i]$ is connected for each $i \in \{1, 2, \ldots, k\}$. Notice that there are only $O(|E|^{k-1})$ possible sets of edges to be deleted. Thus, we enumerate all possible sets of $k-1$ edges, and check whether each set results in a feasible solution. This yields a polynomial-time algorithm for trees when k is fixed. ☐

In the remainder of this section, we thus assume that k is not fixed and is part of the input. Theorem 3 implies that the problem does not admit even a pseudo-polynomial-time algorithm (i.e., an algorithm whose running time is polynomial in $|V|, |E|$, and $\max_{v \in V} w(v)$) for trees unless $\mathbf{P} = \mathbf{NP}$. We thus consider subclasses of trees (more specifically, stars and paths), and/or assume that $|\mathcal{C}|$ is a fixed constant; note that however k is not fixed.

4.1 Polynomial-Time Algorithm for Stars

As the first polynomial-time solvable case, we deal with stars in this subsection. We note that neither $|\mathcal{C}|$ nor k is fixed in the following theorem.

Theorem 5. *The gerrymandering problem is solvable in polynomial time for stars.*

We give such an algorithm as a proof of Theorem 5. Suppose in this subsection that a given graph $G = (V, E)$ is a star having n vertices, whose center vertex is r. For each candidate $q \in \mathcal{C}$, let $L(q) = \{v \in V \setminus \{r\} : c(v) = q\}$. Consider any connected partition $V_1, V_2, \ldots, V_k$ of G; we assume without loss of generality that $r \in V_k$ always holds in this subsection. Then, we know that V_i consists of a single vertex v for each $i \in \{1, 2, \ldots, k-1\}$; and hence $\mathsf{top}(V_i)$ has only one top candidate $c(v)$, that is, $\mathsf{top}(V_i) = \{c(v)\}$. Therefore, for the given partition, we can compute the number of constituencies where the target candidate p wins by checking (i) whether $\mathsf{top}(V_k) = \{p\}$ or not, and (ii) the number of vertices v in $V \setminus V_k$ such that $c(v) = p$, that is, $|L(p) \setminus V_k|$.

Based on (i) and (ii), we now classify the feasible solutions as follows: for a candidate $q^* \in \mathcal{C}$ and an integer $x \in \{1, 2, \ldots, |L(p)|\}$, a feasible solution $V_1, V_2, \ldots, V_k$ to the gerrymandering problem is called a *(q^*, x)-partition* of G if the following holds:

- if $q^* = p$, then $\mathsf{top}(V_k) = \{p\}$ and $|L(p) \setminus V_k| = x$; otherwise $\mathsf{top}(V_k) \ni q^*$ and $|L(p) \setminus V_k| = x + 1$ (that is, p wins alone in exactly $x + 1$ constituencies);
- each candidate $q \in \mathcal{C} \setminus \{p\}$ wins in at most x constituencies.

In this subsection, we will construct a polynomial-time algorithm to check whether there exists a (q^*, x)-partition of G for a given pair of a candidate $q^* \in \mathcal{C}$ and an integer $x \in \{1, 2, \ldots, |L(p)|\}$. Since $|\mathcal{C}| \leq n$ and $|L(p)| \leq n$, by applying this algorithm to all pairs (q^*, x) we can solve the gerrymandering problem in polynomial time.

From now on, we fix a candidate $q^* \in \mathcal{C}$ and an integer $x \in \{1, 2, \ldots, |L(p)|\}$. Our algorithm indeed determines whether there exists a particular (q^*, x)-partition of G, characterized as follows.

Lemma 1. *Assume that G has a (q^*, x)-partition. Then, there exists a (q^*, x)-partition $V_1, V_2, \ldots, V_k$ of G satisfying the following conditions:*

- *$w(u) \geq w(v)$ holds for every pair of vertices $u \in L(q^*) \cap V_k$ and $v \in L(q^*) \setminus V_k$; and*
- *$w(u) \leq w(v)$ holds for every candidate $q \in \mathcal{C} \setminus \{q^*\}$ and every pair of vertices $u \in L(q) \cap V_k$ and $v \in L(q) \setminus V_k$.*

Proof. Let $V_1, V_2, \ldots, V_k$ be any (q^*, x)-partition of G. Assume that there exists a pair of vertices $u \in L(q^*) \cap V_k$ and $v \in L(q^*) \setminus V_k$ such that $w(u) < w(v)$; we assume without loss of generality that $V_1 = \{v\}$. Then, we define $V'_1, V'_2, \ldots, V'_k$, as follows:

$$V'_i \coloneqq \begin{cases} \{u\} & \text{if } i = 1; \\ (V_k \setminus \{u\}) \cup \{v\} & \text{if } i = k; \\ V_i & \text{otherwise.} \end{cases} \tag{1}$$

We now prove that $V'_1, V'_2, \dots, V'_k$ form a (q^*, x)-partition of G. Since $u, v \in V \setminus \{r\}$, we first note that $V'_1, V'_2, \dots, V'_k$ form a connected partition of G. We then note that $\mathsf{top}(V'_k) = \{q^*\}$ holds, since it holds for any candidate $q \in \mathcal{C} \setminus \{q^*\}$ that

$$\sum_{z \in L(q^*) \cap V'_k} w(z) > \sum_{z \in L(q^*) \cap V_k} w(z) \geq \sum_{z \in L(q) \cap V_k} w(z) = \sum_{z \in L(q) \cap V'_k} w(z);$$

the first inequality holds since $V'_k = (V_k \setminus \{u\}) \cup \{v\}$ and $w(v) > w(u)$, and the second inequality holds since $q^* \in \mathsf{top}(V_k)$. We finally prove that p wins alone in exactly $x+1$ constituencies, and any other candidate $q \in \mathcal{C} \setminus \{p\}$ wins in at most x constituencies in the partition. To see this, it suffices to notice that, for all $q \in \mathcal{C}$, we have

$$|\{i \in \{1, 2, \dots, k-1\} : \mathsf{top}(V'_i) = \{q\}\}| = |\{i \in \{1, 2, \dots, k-1\} : \mathsf{top}(V_i) = \{q\}\}|;$$

recall that $u, v \in L(q^*)$ and hence $c(u) = c(v) = q^*$. In this way, we conclude that $V'_1, V'_2, \dots, V'_k$ form a (q^*, x)-partition of G. By repeatedly applying this operation, we obtain a (q^*, x)-partition of G that satisfies the first condition of the lemma.

We next consider any (q^*, x)-partition $V_1, V_2, \dots, V_k$ of G satisfying the first condition of the lemma. Assume that there exist a candidate $q \in \mathcal{C} \setminus \{q^*\}$ and a pair of vertices $u \in L(q) \cap V_k$ and $v \in L(q) \setminus V_k$ such that $w(u) > w(v)$; we assume without loss of generality that $V_1 = \{v\}$. Then, we define $V'_1, V'_2, \dots, V'_k$ by (1). We note that $\mathsf{top}(V'_k) = \mathsf{top}(V_k) \setminus \{q\}$, since we have

$$\sum_{z \in L(q) \cap V'_k} w(z) < \sum_{z \in L(q) \cap V_k} w(z) \leq \sum_{z \in L(q^*) \cap V_k} w(z) = \sum_{z \in L(q^*) \cap V'_k} w(z).$$

Therefore, if $q^* = p$ and hence $\mathsf{top}(V_k) = \{p\}$, then $\mathsf{top}(V'_k) = \{p\}$ holds; and if $q^* \neq p$ and hence $q^* \in \mathsf{top}(V_k)$, then $q^* \in \mathsf{top}(V'_k)$ holds. Then, by the same arguments above for the first condition, we conclude that $V'_1, V'_2, \dots, V'_k$ form a (q^*, x)-partition of G. By repeatedly applying this operation, we obtain a (q^*, x)-partition of G that satisfies both first and second conditions of the lemma. $\square$

We here give a precise description of our algorithm to determine whether there exists a (q^*, x)-partition of a star G satisfying the conditions in Lemma 1. For each $q \in \mathcal{C}$, we denote $L(q) = \{v^q_1, v^q_2, \dots, v^q_{|L(q)|}\}$ and assume that

- $w(v^q_1) \geq w(v^q_2) \geq \cdots \geq w(v^q_{|L(q)|})$ if $q = q^*$; and
- $w(v^q_1) \leq w(v^q_2) \leq \cdots \leq w(v^q_{|L(q)|})$ if $q \neq q^*$.

Since $G = (V, E)$ is a star, a connected partition of G is determined by a subset V_k of V such that $r \in V_k$. Our algorithm tries to construct a subset V_k of V that yields a (q^*, x)-partition of G satisfying the conditions in Lemma 1; if we fail to construct such a subset V_k, then Lemma 1 ensures that there is no (q^*, x)-partition of G.

We first decide the vertices in $V_k \cap L(p)$ for the target candidate p. Recall that p wins in exactly $x+1$ constituencies in any (q^*, x)-partition of G. Then, the number of vertices in $L(p) \setminus V_k$ can be represented by $\alpha(p)$, defined as follows:

$$\alpha(p) \coloneqq \begin{cases} x & \text{if } p = q^*; \\ x+1 & \text{otherwise.} \end{cases}$$

By Lemma 1, we then obtain that

$$V_k \cap L(p) = \{v^p_1, v^p_2, \dots, v^p_{|L(p)| - \alpha(p)}\}. \tag{2}$$

When $q^* \neq p$, we guess the number of vertices in $L(q^*) \setminus V_k$. That is, for $\alpha(q^*) = 1, 2, \dots, \min\{x, |L(q^*)|\}$, we try to find a (q^*, x)-partition of G under the assumption that $|L(q^*) \setminus V_k| = \alpha(q^*)$. By Lemma 1, we obtain that

$$V_k \cap L(q^*) = \{v^{q^*}_1, v^{q^*}_2, \dots, v^{q^*}_{|L(q^*)| - \alpha(q^*)}\}. \tag{3}$$

We then decide the vertices in $V_k \cap L(q)$ for each candidate $q \in \mathcal{C} \setminus \{p, q^*\}$. By (2) and (3), we can define

$$W^{q^*} := \begin{cases} \sum_{u \in V_k \cap L(q^*)} w(u) + w(r) & \text{if } c(r) = q^*; \\ \sum_{u \in V_k \cap L(q^*)} w(u) & \text{otherwise.} \end{cases}$$

For $q \in \mathcal{C} \setminus \{p, q^*\}$ and for $\ell \in \{1, 2, \ldots, |L(q)|\}$, define

$$W_\ell^q := \begin{cases} \sum_{i=1}^{\ell} w(v_i^q) + w(r) & \text{if } c(r) = q; \\ \sum_{i=1}^{\ell} w(v_i^q) & \text{otherwise.} \end{cases}$$

For each $q \in \mathcal{C} \setminus \{p, q^*\}$, let $\beta(q)$ be a minimum non-negative integer such that

- $W^q_{|L(q)|-\beta(q)} < W^{q^*}$ if $q^* = p$;
- $W^q_{|L(q)|-\beta(q)} \leq W^{q^*}$ if $q^* \neq p$,

where we denote $\beta(q) = +\infty$ if such $\beta(q)$ does not exist. Notice that $\beta(q)$ represents the *minimum* number of vertices that have to be contained in $L(q) \setminus V_k$ so that $\mathsf{top}(V_k)$ satisfies the requirement.

Recall that each candidate $q \in \mathcal{C} \setminus \{p, q^*\}$ can win in at most x constituencies in any (q^*, x)-partition of G. Thus, if $\beta(q) \geq x+1$ for some $q \in \mathcal{C} \setminus \{p, q^*\}$, then we can immediately conclude that G has no (q^*, x)-partition. We also observe that, if $\beta(q) = x$ and $W^q_{|L(q)|-\beta(q)} = W^{q^*}$ for some $q \in \mathcal{C} \setminus \{p, q^*\}$, then q wins in $x+1$ constituencies, and hence G has no (q^*, x)-partition. If neither of the above conditions holds, then for $q \in \mathcal{C} \setminus \{p, q^*\}$, $|V_k \cap L(q)|$ can take an arbitrary integer satisfying $\beta(q) \leq |V_k \cap L(q)| \leq \min\{x, L(q)\}$.

Therefore, the existence of a desired (q^*, x)-partition is equivalent to

$$\sum_{q \in \mathcal{C} \setminus \{p\}} \beta(q) \leq k - 1 - \alpha(p) \leq \sum_{q \in \mathcal{C} \setminus \{p\}} \min\{x, L(q)\}$$

if $q^* = p$, and

$$\sum_{q \in \mathcal{C} \setminus \{p, q^*\}} \beta(q) \leq k - 1 - \alpha(p) - \alpha(q^*) \leq \sum_{q \in \mathcal{C} \setminus \{p, q^*\}} \min\{x, L(q)\}$$

if $q^* \neq p$.

Since the number of choices of $\alpha(q^*)$ is at most $\min\{x, L(q^*)\}$, the algorithm above runs in polynomial time for each candidate $q^* \in \mathcal{C}$ and each integer $x \in \{1, 2, \ldots, |L(p)|\}$. Therefore, we obtain a polynomial-time algorithm for stars.

4.2 Polynomial-Time Algorithm for Paths with Fixed $|\mathcal{C}|$

As the second polynomial-time solvable case, we consider paths when $|\mathcal{C}|$ is fixed. We note that the problem is not so straightforward even for paths: Recall the example in Figure 2, where the vertex u should form a singleton even if p can win alone in $\{u, v\}$; greedily enlarging the constituency having a vertex z with $c(z) = p$ does not always yield a feasible solution. We thus construct a dynamic programming algorithm, and obtain the following theorem.

Theorem 6. *The gerrymandering problem is solvable in polynomial time for paths when $|\mathcal{C}|$ is a fixed constant.*

We give such an algorithm as a proof of Theorem 6. Suppose in this subsection that a given graph G is a path with n vertices and $|\mathcal{C}|$ is a fixed constant; for notational convenience, we assume that the path is drawn from left to right. Roughly speaking, our algorithm employs a dynamic programming method, which computes and extends partial solutions for sub-paths from left to right by keeping the frontier (i.e., the rightmost constituency) of a partial solution together with the information on the way how the candidates in $\mathcal{C}$ win in the partial solution.

We now define partial solutions for sub-paths. Let $v_1, v_2, \dots, v_n$ be the vertices in G ordered from left to right. For a pair of integers i, j, $1 \le i \le j \le n$, we denote by $G_{i,j}$ the sub-path of G consisting of vertices $v_i, v_{i+1}, \dots, v_j$; note that $G_{i,i}$ consists of a single vertex v_i. We call any mapping $t \colon 2^{\mathcal{C}} \to \{0, 1, \dots, k\}$ a *top configuration*, which will characterize how the candidates in $\mathcal{C}$ win in a partial solution. We note that there are only a polynomial number of distinct top configurations t; more specifically, it is $O(k^{2^{|\mathcal{C}|}}) = O(n^{2^{|\mathcal{C}|}})$. For a pair of integers i, j, $1 \le i \le j \le n$, and a top configuration t, we call a partition $V_1, V_2, \dots, V_{k'}$ of $V(G_{1,j})$ an $(i, j; t)$-*partition* of $G_{1,j}$ if the following four conditions hold:

1. $k' = \sum_{X \subseteq \mathcal{C}} t(X)$;
2. $V_{k'} = \{v_i, v_{i+1}, \dots, v_j\}$;
3. $G[V_z]$ is connected for each $z \in \{1, 2, \dots, k' - 1\}$; and
4. $|\{z \in \{1, 2, \dots, k'\} : \mathsf{top}(V_z) = X\}| = t(X)$ for all $X \subseteq \mathcal{C}$, that is, $t(X)$ is the number of districts in which the set of top candidates is exactly X.

We regard $(i, j; t)$-partitions of $G_{1,j}$ as partial solutions of $G_{1,j}$, and call the rightmost constituency $G_{i,j} = G[V_{k'}]$ the *frontier* of an $(i, j; t)$-partition. We then define the following function: for integers i, j, $1 \le i \le j \le n$, and a top configuration $t \colon 2^{\mathcal{C}} \to \{0, 1, \dots, k\}$, let

$$\phi(i, j; t) \coloneqq \begin{cases} \mathsf{yes} & \text{if } G_{1,j} \text{ has an } (i, j; t)\text{-partition;} \\ \mathsf{no} & \text{otherwise.} \end{cases}$$

Then, there is a feasible solution to a given instance of the gerrymandering problem if and only if there exists a pair of $i \in \{1, 2, \dots, n\}$ and a top configuration t such that $\phi(i, n; t) = \mathsf{yes}$, $\sum_{X \subseteq \mathcal{C}} t(X) = k$, and $t(\{p\}) > \sum_{X \subseteq \mathcal{C} \colon q \in X} t(X)$ for all $q \in \mathcal{C} \setminus \{p\}$. In our algorithm for the gerrymandering problem, we compute $\phi(i, n; t)$ for all i and t, and then check whether there exist i and t satisfying the above conditions.

In order to compute $\phi(i, n; t)$, our algorithm computes $\phi(i, j; t)$ for all possible triples (i, j, t) from left to right of a given path G as follows.

Initialization. We first compute $\phi(i, j; t)$ for all (i, j, t) such that $i = 1$. Notice that $V(G_{1,j})$ itself is the frontier when $i = 1$. Therefore, $\phi(1, j, t) = \mathsf{yes}$, $1 \le j \le n$, holds if and only if the top configuration $t \colon 2^{\mathcal{C}} \to \{0, 1, \dots, k\}$ satisfies

$$t(X) = \begin{cases} 1 & \text{if } X = \mathsf{top}(V(G_{1,j})); \\ 0 & \text{otherwise.} \end{cases}$$

Update. The case where $i \ge 2$ can be computed as follows. For two integers i, j, $1 \le i \le j \le n$, and a top configuration t, we have $\phi(i, j; t) = \bigvee \phi(h, i-1; t')$, where the OR operation is taken over all integers h, $1 \le h \le i - 1$, and the top configuration t' defined as follows: for each $X \subseteq \mathcal{C}$,

$$t'(X) \coloneqq \begin{cases} t(X) - 1 & \text{if } X = \mathsf{top}(V(G_{i,j})); \\ t(X) & \text{otherwise.} \end{cases}$$

Recall that there are $O(k^{2^{|\mathcal{C}|}}) = O(n^{2^{|\mathcal{C}|}})$ distinct top configurations t, and $|\mathcal{C}|$ is fixed in this subsection. Therefore, our algorithm above runs in polynomial time. This completes the proof of Theorem 6.

4.3 Pseudo-Polynomial-Time Algorithm for Trees with Fixed $|\mathcal{C}|$

Recall again that the gerrymandering problem does not admit even a pseudo-polynomial-time algorithm for trees in general unless $\mathbf{P} = \mathbf{NP}$ (Theorem 3). However, if $|\mathcal{C}|$ is a fixed constant, we have the following theorem for trees.

Theorem 7. *The gerrymandering problem is solvable in pseudo-polynomial time for trees when $|\mathcal{C}|$ is a fixed constant.*

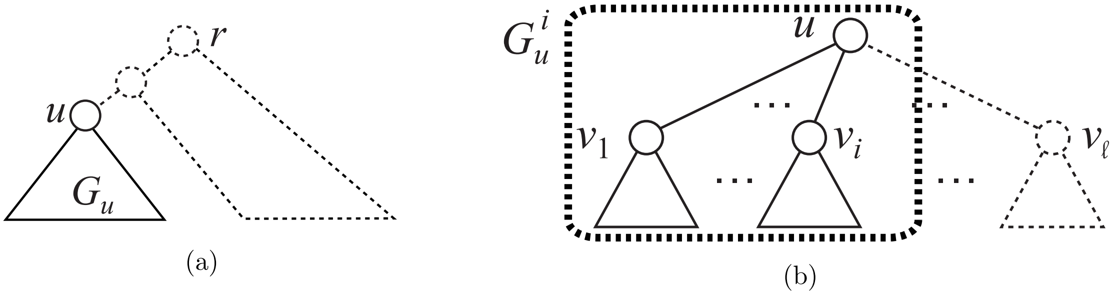


Figure 5: (a) Subtree G_u in a whole tree G and (b) subtree G_u^i in G_u.

We give such an algorithm as a proof of Theorem 7. Suppose in this subsection that a given graph G is a tree with n vertices and $|\mathcal{C}|$ is a fixed constant. We choose an arbitrary vertex r in $V(G)$ as the root of G, and regard G as a rooted tree. Similarly to paths, our algorithm employs a dynamic programming method, which computes and extends partial solutions for subtrees from the leaves to the root of G. However, in contrast to the path case, we need a special care when we keep the frontier (i.e., the constituency containing the root of each subtree) in a partial solution. Although it sufficed to specify only two endpoints of the frontier (i.e., two integers i and j) in the path case, the tree case may require us to specify $O(n)$ endpoints of the frontier, which would result in an exponential-time algorithm. We thus characterize the frontier of a partial solution only by the weight that each candidate obtains; this will yield a pseudo-polynomial-time algorithm for trees.

We now define partial solutions for subtrees. For each vertex u in $V(G)$, let G_u be the subtree of G that is rooted at u and is induced by u and all descendants of u on G. (See Figure 5(a).) Denote the children of u by $v_1, v_2, \ldots, v_\ell$, ordered arbitrarily. For each $i \in \{1, 2, \ldots, \ell\}$, we denote by G_u^i the subtree of G induced by $\{u\} \cup V(G_{v_1}) \cup V(G_{v_2}) \cup \cdots \cup V(G_{v_i})$. For example, in Figure 5(b), the subtree G_u^i is surrounded by a thick dotted rectangle. For notational convenience, we denote by G_u^0 the tree consisting of a single vertex u. Then, $G_u = G_u^0$ for each leaf u of G. Let $W := \sum_{u \in V(G)} w(u)$, and let $\mathbb{Z}_W := \{0, 1, \ldots, W\}$. We call a vector $\vec{x} \in \mathbb{Z}_W^{\mathcal{C}}$ a *weight configuration*, which characterizes the weight that each candidate in $\mathcal{C}$ obtains in the frontier of a partial solution. For a subtree G_u^i, a top configuration $t \colon 2^{\mathcal{C}} \to \{0, 1, \ldots, k\}$, and a weight configuration $\vec{x} \in \mathbb{Z}_W^{\mathcal{C}}$, we call a partition $V_1, V_2, \ldots, V_{k'}$ of $V(G_u^i)$ a *$(t, \vec{x})$-partition* of G_u^i if the following four conditions hold:

1. $k' - 1 = \sum_{X \subseteq \mathcal{C}} t(X)$;
2. $G[V_z]$ is connected for each $z \in \{1, 2, \ldots, k'\}$, and $u \in V_{k'}$;
3. $|\{z \in \{1, 2, \ldots, k'-1\} : \mathsf{top}(V_z) = X\}| = t(X)$ for all $X \subseteq \mathcal{C}$; and
4. $\sum_{v \in V_{k'} \colon c(v) = q} w(v) = \vec{x}(q)$ for all $q \in \mathcal{C}$.

We regard $(t, \vec{x})$-partitions of G_u^i as partial solutions of G_u^i, and call the constituency $G[V_{k'}]$ containing the root u of G_u^i the *frontier* of a $(t, \vec{x})$-partition. Note that, by the condition 1 of the definition above, k' is automatically determined when t is fixed. Note also that the condition 3 of the definition above means that $t(X)$ is the number of districts in which the set of top candidates is exactly X, where the set $\mathsf{top}(V_{k'})$ of top candidates in the frontier is not counted, since this frontier $G[V_{k'}]$ may be extended later. However, $\mathsf{top}(V_{k'}) = \arg\max_{q \in \mathcal{C}} \{\vec{x}(q)\}$ holds, and hence $\mathsf{top}(V_{k'})$ can be computed only from $\vec{x}$. For a top configuration t and each $X \subseteq \mathcal{C}$, we define

$$t_{\vec{x}}(X) := \begin{cases} t(X) + 1 & \text{if } X = \arg\max_{q \in \mathcal{C}} \{\vec{x}(q)\}; \\ t(X) & \text{otherwise.} \end{cases}$$

We then define the following function: For a subtree G_u^i, a top configuration $t \colon 2^{\mathcal{C}} \to \{0, 1, \ldots, k\}$, and a weight configuration $\vec{x} \in \mathbb{Z}_W^{\mathcal{C}}$, we let

$$\varphi(G_u^i; t, \vec{x}) := \begin{cases} \mathsf{yes} & \text{if } G_u^i \text{ has a } (t, \vec{x})\text{-partition;} \\ \mathsf{no} & \text{otherwise.} \end{cases}$$

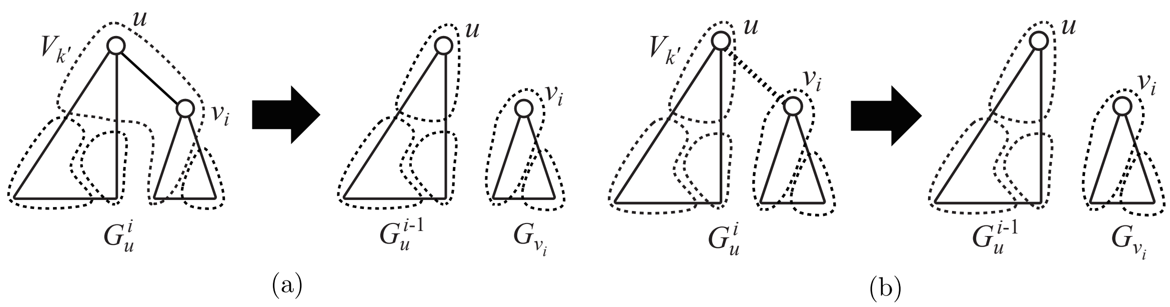


Figure 6: $(t, \vec{x})$-partitions of a subtree G_u^i, and their restrictions to subtrees G_u^{i-1} and G_{v_i}.

Then, there is a feasible solution to a given instance of the gerrymandering problem if and only if there exists a pair of a top configuration t and a weight configuration $\vec{x}$ such that $\varphi(G; t, \vec{x}) = \mathsf{yes}$, $\sum_{X \subseteq \mathcal{C}} t_{\vec{x}}(X) = k$, and $t_{\vec{x}}(\{p\}) > \sum_{X \subseteq \mathcal{C}:\, q \in X} t_{\vec{x}}(X)$ for all $q \in \mathcal{C} \setminus \{p\}$. In our algorithm for the gerrymandering problem, we compute $\varphi(G; t, \vec{x})$ for all t and $\vec{x}$, and then check whether there exist t and $\vec{x}$ satisfying the above conditions.

For a given tree G, in order to compute $\varphi(G; t, \vec{x})$, our algorithm computes $\varphi(G_u^i; t, \vec{x})$ for all possible triples $(G_u^i, t, \vec{x})$ from the leaves to the root r as follows.

Initialization. We first compute $\varphi(G_u^0; t, \vec{x})$ for all vertices $u \in V(G)$ (including internal vertices in G). Recall that G_u^0 consists of a single vertex u. Therefore, $\varphi(G_u^0; t, \vec{x}) = \mathsf{yes}$ holds if and only if $t(X) = 0$ for all $X \subseteq \mathcal{C}$ and $\vec{x}$ satisfies

$$\vec{x}(q) = \begin{cases} w(u) & \text{if } q = c(u); \\ 0 & \text{otherwise} \end{cases}$$

for each $q \in \mathcal{C}$. Notice that we have computed $\varphi(G_u; t, \vec{x})$ for all leaves of G, since $G_u = G_u^0$ if u is a leaf.

Update. We now consider the case where $i \geq 1$. To compute $\varphi(G_u^i; t, \vec{x})$, we classify the partial solutions of G_u^i into the following two groups (a) and (b).

(a) The vertices u and v_i are contained in the same connected component. (See also Figure 6(a).)

In this case, the edge uv_i is not deleted, and the frontier in a $(t, \vec{x})$-partition of G_u^i can be obtained by merging the frontier in a $(t', \vec{y})$-partition of G_u^{i-1} with the frontier in a $(t'', \vec{z})$-partition of G_{v_i}. Thus, we define

$$\varphi^a(G_u^i; t, \vec{x}) \coloneqq \bigvee \left(\varphi(G_u^{i-1}; t', \vec{y}) \wedge \varphi(G_{v_i}; t'', \vec{z}) \right),$$

where the OR operation $\bigvee$ is taken over all top configurations $t', t'' \colon 2^{\mathcal{C}} \to \{0, 1, \ldots, k\}$ and all weight configurations $\vec{y}, \vec{z} \in \mathbb{Z}_W^{\mathcal{C}}$ such that $t'(X) + t''(X) = t(X)$ for each $X \subseteq \mathcal{C}$, and $\vec{y}(q) + \vec{z}(q) = \vec{x}(q)$ for each $q \in \mathcal{C}$.

(b) The vertices u and v_i are not contained in the same connected component. (See also Figure 6(b).)

In this case, the edge uv_i is deleted, and the frontier in a $(t, \vec{x})$-partition of G_u^i is the frontier in a $(t', \vec{x})$-partition of G_u^{i-1}. Note that the frontier $V_{k''}$ in a $(t'', \vec{z})$-partition of G_{v_i} is merely a connected component in the $(t, \vec{x})$-partition of G_u^i. Thus, we can compute $\mathsf{top}(V_{k''})$, and have to take the top candidates in $V_{k''}$ into account. Therefore, we define

$$\varphi^b(G_u^i; t, \vec{x}) \coloneqq \bigvee \left(\varphi(G_u^{i-1}; t', \vec{x}) \wedge \varphi(G_{v_i}; t'', \vec{z}) \right),$$

where the OR operation $\bigvee$ is taken over all top configurations $t', t'' \colon 2^{\mathcal{C}} \to \{0, 1, \ldots, k\}$ and all weight configurations $\vec{z} \in \mathbb{Z}_W^{\mathcal{C}}$ such that $t'(X) + t''_{\vec{z}}(X) = t(X)$ for each $X \subseteq \mathcal{C}$.

Then, $\varphi(G_u^i; t, \vec{x}) = \varphi^a(G_u^i; t, \vec{x}) \vee \varphi^b(G_u^i; t, \vec{x})$. Recall that there are $O(k^{2^{|\mathcal{C}|}})$ distinct top configurations t, and notice that $|\mathbb{Z}_W^{\mathcal{C}}| = O(W^{|\mathcal{C}|})$. Since $|\mathcal{C}|$ is fixed in this subsection, our algorithm above computes $\varphi(G_u^i; t, \vec{x})$ for all possible triples $(G_u^i, t, \vec{x})$ in pseudo-polynomial time. Furthermore, in pseudo-polynomial time, we can check whether there exists a pair of a top configuration t and a weight configuration $\vec{x}$ such that $\varphi(G; t, \vec{x}) = \mathsf{yes}$, $\sum_{X \subseteq \mathcal{C}} t_{\vec{x}}(X) = k$, and $t_{\vec{x}}(\{p\}) > \sum_{X \subseteq \mathcal{C}:\, q \in X} t_{\vec{x}}(X)$ for all $q \in \mathcal{C} \setminus \{p\}$ by enumerating all possible pairs. This completes the proof of Theorem 7.

5 Algorithms for Complete Graphs

In this section, we consider complete graphs. Recall that the gerrymandering problem is **NP**-complete for complete graphs even if $k = |\mathcal{C}| = 2$ (Theorem 1). In this section, for each candidate $q \in \mathcal{C}$, we define $T(q) \coloneqq \{v \in V : c(v) = q\}$.

We give the following theorem for complete graphs and $k = 2$; note that $|\mathcal{C}|$ is not necessarily fixed.

Theorem 8. *The gerrymandering problem is solvable in pseudo-polynomial time for complete graphs and $k = 2$.*

Proof. Since $G = (V, E)$ is a complete graph, any vertex subset $U \subseteq V$ induces a connected subgraph. Furthermore, since $k = 2$, the target candidate p must win alone in both constituencies $G[V_1]$ and $G[V_2]$ in any feasible solution V_1, V_2. Thus, the problem for complete graphs and $k = 2$ can be rephrased as follows: For each pair of nonnegative integers W_1 and W_2 such that $W_1 + W_2 = \sum_{v \in T(p)} w(v)$, we wish to determine whether each vertex set $T(q)$, $q \in \mathcal{C}$, can be partitioned into two subsets T_q^1 and T_q^2 such that

- if $q = p$, then $\sum_{v \in T_p^1} w(v) = W_1$ and $\sum_{v \in T_p^2} w(v) = W_2$; and
- if $q \in \mathcal{C} \setminus \{p\}$, then $\sum_{v \in T_q^1} w(v) < W_1$ and $\sum_{v \in T_q^2} w(v) < W_2$.

If there is a pair of W_1 and W_2 such that desired partitions T_q^1, T_q^2 of $T(q)$ exist for all $q \in \mathcal{C}$, then there is a feasible solution to the gerrymandering problem. For each $q \in \mathcal{C}$, the existence of such a partition of $T(q)$ can be checked by a pseudo-polynomial-time algorithm for the subset sum problem [14]. ☐

Finally, we show an interesting contrast on complete graphs: the problem is solvable in polynomial time for complete graphs and any $k \geq 3$. The feasibility of the gerrymandering problem for such a case can be characterized by the following (4); furthermore, it yields a polynomial-time algorithm.

Theorem 9. *The gerrymandering problem is solvable in polynomial time for complete graphs and any $k \geq 3$. In particular, there exists a feasible solution to such an instance if and only if it holds that*

$$|T(p)| + \sum_{q \in \mathcal{C} \setminus \{p\}} \min\{|T(q)|, |T(p)| - 1\} \geq k. \tag{4}$$

Proof. It suffices to prove that there exists a feasible solution for a complete graph G and any $k \geq 3$ if and only if (4) holds, since we can check in polynomial time whether (4) holds or not.

We first prove the necessity. Assume that there exists a feasible solution $V_1, V_2, \ldots, V_k$ to the gerrymandering problem. We define $\alpha \coloneqq |\{i \in \{1, 2, \ldots, k\} : \{p\} = \mathsf{top}(V_i)\}|$, and $\beta(q) \coloneqq |\{i \in \{1, 2, \ldots, k\} : q \in \mathsf{top}(V_i)\}|$ for each $q \in \mathcal{C} \setminus \{p\}$. Then, we have $\alpha \leq |T(p)|$ and $\beta(q) \leq |T(q)|$ for each $q \in \mathcal{C} \setminus \{p\}$. Furthermore, since $V_1, V_2, \ldots, V_k$ is a feasible solution of the gerrymandering problem, $\beta(q) \leq |T(p)| - 1$ holds for each $q \in \mathcal{C} \setminus \{p\}$. Thus, we have

$$\alpha + \sum_{q \in \mathcal{C} \setminus \{p\}} \beta(q) \leq |T(p)| + \sum_{q \in \mathcal{C} \setminus \{p\}} \min\{|T(q)|, |T(p)| - 1\}. \tag{5}$$

On the other hand, we have

$$\alpha + \sum_{q \in \mathcal{C} \setminus \{p\}} \beta(q) = \alpha + \sum_{i=1}^{k} |\mathsf{top}(V_i) \setminus \{p\}| \geq \alpha + |\{i \in \{1, 2, \ldots, k\} : \{p\} \neq \mathsf{top}(V_i)\}| = k. \tag{6}$$

Thus, (4) follows from (5) and (6).

We next show the sufficiency. Assume that (4) holds.

We first consider the case where $|T(p)| \geq k$. Let $X_1, X_2, \ldots, X_{k-1}$ be an arbitrary partition of $T(p)$. Then, we define $V_i \coloneqq X_i$ for each $i \in \{1, 2, \ldots, k-1\}$ and $V_k \coloneqq V \setminus T(p)$. The definition of $V_1, V_2, \ldots, V_k$ implies that

- $|\{i \in \{1, 2, \ldots, k\} : \{p\} = \mathsf{top}(V_i)\}| = k - 1$, and

- $|\{i \in \{1, 2, \ldots, k\} : q \in \mathsf{top}(V_i)\}| \leq 1$ for all $q \in \mathcal{C} \setminus \{p\}$.

Since $k \geq 3$ and hence $k - 1 > 1$, $V_1, V_2, \ldots, V_k$ forms a feasible solution of the gerrymandering problem.

Next we consider the case where $|T(p)| < k$. We denote $\mathcal{C} \setminus \{p\} = \{q_1, q_2, \ldots, q_\ell\}$; the candidates are ordered arbitrarily. Let $\ell' \in \{1, 2, \ldots, \ell\}$ be the integer such that

$$|T(p)| + \sum_{j=1}^{\ell'-1} \min\{|T(q_j)|, |T(p)| - 1\} < k, \quad \text{and}$$

$$|T(p)| + \sum_{j=1}^{\ell'} \min\{|T(q_j)|, |T(p)| - 1\} \geq k.$$

Notice that (4) and $|T(p)| < k$ imply the existence of such an integer ℓ'. For each integer $j \in \{1, 2, \ldots, \ell' - 1\}$, we define $\gamma_j \coloneqq \min\{|T(q_j)|, |T(p)| - 1\}$. Furthermore, we define $\gamma_{\ell'}$ by

$$\gamma_{\ell'} \coloneqq k - |T(p)| - \sum_{j=1}^{\ell'-1} \min\{|T(q_j)|, |T(p)| - 1\} \leq \min\{|T(q_{\ell'})|, |T(p)| - 1\}.$$

Let $X_1, X_2, \ldots, X_{|T(p)|}$ be the partition of $T(p)$ into singletons. For each $j \in \{1, 2, \ldots, \ell' - 1\}$, let $Y_1^j, Y_2^j, \ldots, Y_{\gamma_j}^j$ be an arbitrary partition of $T(q_j)$. Furthermore, let $Y_1^{\ell'}, Y_2^{\ell'}, \ldots, Y_{\gamma_{\ell'}}^{\ell'}$ be an arbitrary partition of $\{v \in V : c(v) \notin \{p, q_1, q_2, \ldots, q_{\ell'-1}\}\}$. Then, we define a partition $(V_1, V_2, \ldots, V_k)$ of V by

$$(X_1, X_2, \ldots, X_{|T(p)|}, Y_1^1, Y_2^1, \ldots, Y_{\gamma_1}^1, \ldots, Y_1^{\ell'}, Y_2^{\ell'}, \ldots, Y_{\gamma_{\ell'}}^{\ell'}).$$

The definition of $V_1, V_2, \ldots, V_k$ implies that

- $|\{i \in \{1, 2, \ldots, k\} : \{p\} = \mathsf{top}(V_i)\}| = |T(p)|$,
- $|\{i \in \{1, 2, \ldots, k\} : q_j \in \mathsf{top}(V_i)\}| = \gamma_j \leq |T(p)| - 1$ for all $j \in \{1, 2, \ldots, \ell' - 1\}$, and
- $|\{i \in \{1, 2, \ldots, k\} : q_j \in \mathsf{top}(V_i)\}| \leq \gamma_{\ell'} \leq |T(p)| - 1$ for all $j \in \{\ell', \ell' + 1, \ldots, \ell\}$.

Thus, $V_1, V_2, \ldots, V_k$ form a feasible solution of the gerrymandering problem. ◻

6 Conclusion

In this paper, we gave several hardness results and polynomial-time algorithms for gerrymandering over graphs. The main open problem left in this paper is to settle the complexity status for paths when the number of candidates is not fixed. The polynomial-time solvability for trees also remains open when the number of candidates is fixed, whereas we give a pseudo-polynomial-time algorithm for this case. The complexity for trees of diameter three also remains unclear. The problem under other voting rules should also be investigated. In particular, it is natural to consider partitions into (almost) equal sized parts as in [8]. Parameterized complexity of the problem is also a natural direction of further research.

Acknowledgements

The first author was partially supported by JST CREST Grant Number JPMJCR1402, and JSPS KAKENHI Grant Numbers JP16K00004, JP18H04091, JP19K11814, and JP20H05793. The second author was partially supported by JST, PRESTO Grant Number JPMJPR1753, Japan. The third author was partially supported by JST ACT-I Grant Number JPMJPR17UB, and JSPS KAKENHI Grant Numbers JP16K16010, JP18H05291, JP19H05485, JP20K11692, and JP20H05795. The fourth author was partially supported by JSPS KAKENHI Grant Numbers JP15K00009, JP20K11670, JP20H05795, JST CREST Grant Number JPMJCR1402, and Kayamori Foundation of Informational Science Advancement.

References

[1] N. Apollonio, R. Becker, I. Lari, F. Ricca, and B. Simeone. Bicolored graph partitioning, or: gerrymandering at its worst. *Discrete Applied Mathematics*, 157(17):3601–3614, 2009.

[2] Y. Bachrach, O. Lev, Y. Lewenberg, and Y. Zick. Misrepresentation in district voting. In S. Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 81–87. IJCAI/AAAI Press, 2016.

[3] N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theor. Comput. Sci.*, 410(52):5425–5442, 2009.

[4] A. Borodin, O. Lev, N. Shah, and T. Strangway. Big city vs. the great outdoors: Voter distribution and how it affects gerrymandering. In J. Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 98–104. ijcai.org, 2018.

[5] B. Bozkaya, E. Erkut, and G. Laporte. A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1):12–26, 2003.

[6] V. Cohen-Addad, P. N. Klein, and N. E. Young. Balanced centroidal power diagrams for redistricting. In F. B. Kashani, E. G. Hoel, R. H. Güting, R. Tamassia, and L. Xiong, editors, *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2018, Seattle, WA, USA, November 06-09, 2018*, pages 389–396. ACM, 2018.

[7] A. Cohen-Zemach, Y. Lewenberg, and J. S. Rosenschein. Gerrymandering over graphs. In E. André, S. Koenig, M. Dastani, and G. Sukthankar, editors, *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 274–282. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018.

[8] M. Dyer and A. Frieze. On the complexity of partitioning graphs into connected subgraphs. *Discrete Applied Mathematics*, 10(2):139–153, 1985.

[9] G. Erdélyi, M. R. Fellows, J. Rothe, and L. Schend. Control complexity in bucklin and fallback voting: A theoretical analysis. *J. Comput. Syst. Sci.*, 81(4):632–660, 2015.

[10] G. Erdélyi, E. Hemaspaandra, and L. A. Hemaspaandra. More natural models of electoral control by partition. In T. Walsh, editor, *Algorithmic Decision Theory - 4th International Conference, ADT 2015, Lexington, KY, USA, September 27-30, 2015, Proceedings*, volume 9346 of *Lecture Notes in Computer Science*, pages 396–413. Springer, 2015.

[11] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and copeland voting computationally resist bribery and constructive control. *J. Artif. Intell. Res.*, 35:275–341, 2009.

[12] P. Faliszewski and J. Rothe. Control and bribery in voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, pages 146–168. Cambridge University Press, 2016.

[13] B. Fleiner, B. Nagy, and A. Tasnádi. Optimal partisan districting on planar geographies. *Central European Journal of Operations Research*, 25(4):879–888, Dec 2017.

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

[15] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artif. Intell.*, 171(5-6):255–285, 2007.

[16] J. J. B. III, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, Jul 1989.

[17] J. J. B. III, C. A. Tovey, and M. A. Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(8):27–40, 1992.

[18] Y. Lewenberg, O. Lev, and J. S. Rosenschein. Divide and conquer: Using geographic manipulation to win district-based elections. In K. Larson, M. Winikoff, S. Das, and E. H. Durfee, editors, *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 624–632. ACM, 2017.

[19] H. Liu, H. Feng, D. Zhu, and J. Luan. Parameterized computational complexity of control problems in voting systems. *Theor. Comput. Sci.*, 410(27–29):2746–2753, 2009.

[20] A. Mehrotra, E. L. Johnson, and G. L. Nemhauser. An optimization based heuristic for political districting. *Management Science*, 44(8):1100–1114, 1998.

[21] C. Puppe and A. Tasnádi. Optimal redistricting under geographical constraints: Why "pack and crack" does not work. *Economics Letters*, 105(1):93–96, 2009.

[22] R. van Bevern, R. Bredereck, J. Chen, V. Froese, R. Niedermeier, and G. J. Woeginger. Network-based vertex dissolution. *SIAM J. Discret. Math.*, 29(2):888–914, 2015.

[23] L. Vanneschi, R. Henriques, and M. Castelli. Multi-objective genetic algorithm with variable neighbourhood search for the electoral redistricting problem. *Swarm and Evolutionary Computation*, 36:37–51, 2017.