Manipulating Districts to Win Elections: Fine-Grained Complexity

Eduard Eiben,[1] **Fedor V. Fomin,**[2] **Fahad Panolan,**[3] **Kirill Simonov**[2]
[1]Department of Computer Science, Royal Holloway, University of London, UK
[2]Department of Informatics, University of Bergen, Norway
[3]Department of Computer Science and Engineering, IIT Hyderabad, India
eduard.eiben@rhul.ac.uk, {fedor.fomin, kirill.simonov}@uib.no, fahad.panolan@gmail.com

Abstract

Gerrymandering is a practice of manipulating district boundaries and locations in order to achieve a political advantage for a particular party. Lewenberg, Lev, and Rosenschein [AAMAS 2017] initiated the algorithmic study of a geographically-based manipulation problem, where voters must vote at the ballot box closest to them. In this variant of gerrymandering, for a given set of possible locations of ballot boxes and known political preferences of n voters, the task is to identify locations for k boxes out of m possible locations to guarantee victory of a certain party in at least ℓ districts. Here integers k and ℓ are some selected parameter.

It is known that the problem is NP-complete already for 4 political parties and prior to our work only heuristic algorithms for this problem were developed. We initiate the rigorous study of the gerrymandering problem from the perspectives of parameterized and fine-grained complexity and provide asymptotically matching lower and upper bounds on its computational complexity. We prove that the problem is W[1]-hard parameterized by $k + n$ and that it does not admit an $f(n, k) \cdot m^{o(\sqrt{k})}$ algorithm for any function f of k and n only, unless the Exponential Time Hypothesis (ETH) fails. Our lower bounds hold already for 2 parties. On the other hand, we give an algorithm that solves the problem for a constant number of parties in time $(m + n)^{\mathcal{O}(\sqrt{k})}$.

Introduction

In 1812, Massachusetts governor Elbridge Gerry immortalized his name by signing a bill that created a mythological salamander shaped district in the Boston area to the benefit of his political party. Since then the term gerrymandering has been used for the practice of establishing a political advantage by manipulating voting district boundaries. Wikipedia article Gerrymandering contains a lot of notable gerrymandering examples in electoral systems of many countries. One of possible responses of such manipulations is a more “rational” definition of voting districts, for example the system where voters should always go to a ballot box (or central area) that is closest to them (Terbush 2013).

However, even in the “rational” geographical settings manipulations are still possible. Lewenberg, Lev, and Rosenschein (2017) considered the variant of the gerrymandering problem in which the agent in charge of the design of voting districts has to follow a clear rule: *all voters vote in the ballot box nearest to them.* As it was shown by Lewenberg, Lev, and Rosenschein (2017), even with clear and rational rules, manipulation is possible. This brought Lewenberg et al. to the following natural question. Is there an efficient algorithmic procedure that would allow an agent, who is obliged to follow the rules, to optimize division for his personal preferred outcome? The good news here is that the existence of such a procedure is highly unlikely, because as it was shown in (Lewenberg, Lev, and Rosenschein 2017), the problem is NP-complete already for 4 political parties.

On the other hand, when the number of districts k is small, the problem is trivially solvable in time $m^k n^{\mathcal{O}(1)}$, where m is the number of all possible locations of ballot boxes and n is the number of voters, by a brute-force enumerating all possible locations for k boxes. Thus in the situation when the number of districts is small, there is an efficient procedure for finding an optimal manipulation. This immediately brings us to the following question which serves as the departure point of our work.

Question 1. *Is it possible to solve the gerrymandering problem faster than the brute-force?*

To answer this question, we study GERRYMANDERING through the lens of Parameterized Complexity, which is a multivariate paradigm of algorithm analysis introduced by Downey and Fellows (1999). In Parameterized Complexity, one measures the performance of algorithms not merely in terms of the size of the input, but also with respect to certain properties of the input or the output (captured by one or several numerical *parameters*, k). This gives rise to two notions of tractability, both of which correspond to polynomial-time tractability in the classical setting, when the parameter is constant. First is the class FPT that contains all problems that can be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$ (for some computable function f), while the (asymptotically less efficient) class XP contains all problems that can be solved in time $n^{f(k)}$. Aside from these complexity classes, we will make use of the complexity class W[1], the class of parameterized problems that can be in time $f(k) \cdot n^{\mathcal{O}(1)}$ reduced to INDEPENDENT SET parameterized by the solution size. FPT≠W[1] is the working hypothesis of Parameterized Complexity and it is widely believed that no W[1]-hard problem is in FPT. Finally, our more fine-grained lower bound result depends on the Exponential Time Hypothesis (ETH) (Impagliazzo and Paturi 2001; Lokshtanov, Marx, and Saurabh 2011), which states that the satisfiability of k-CNF formulas (for $k \geq 3$) is not solvable in subexponential-time $2^{o(n)}$, where n is the number of variables in the formula. We refer to the recent books of Cygan et al. (2015) and Downey and Fellows (2013) for more exposition on these complexity notions.

Contribution Lewenberg, Lev, and Rosenschein (2017) proved that GERRYMANDERING$_{plurality}$ (the Gerrymandering problem when the winner at each district is decided by *plurality* voting rule) is NP-complete, even when the number of candidates $|C|$ is 4. We enhance this intractability result by providing a fine-grained complexity of the problem. Also note that our reduction also implies that the problem is NP-complete for any number $|C| \geq 2$ of candidates.

Theorem 1. *For any number $|C| \geq 2$ of candidates,* GERRYMANDERING$_{plurality}$ *is* W[1]-hard *parameterized by $k + n$. Moreover, there is no algorithm solving* GERRYMANDERING$_{plurality}$ *in time $f(k, n) \cdot m^{o(\sqrt{k})}$ for any computable function f, unless* ETH *fails.*

The tools we used to obtain lower bounds are based on the grid-tiling technique of Marx (2007). We refer to the book of Cygan et al. (2015) for further discussions of subexponential algorithms.

Interestingly, the ETH lower bound from Theorem 1 is asymptotically tight. We give an algorithm whose running time matches our lower bound.

Theorem 2. *For any constant number of candidates and for any voting rule g,* GERRYMANDERING$_g$ *is solvable in time $(m + n)^{\mathcal{O}(\sqrt{k})}$.*

To obtain our algorithm, we use the machinery developed by Marx and Pilipczuk (2015). The main idea is to guess roughly $\sqrt{k}$ many ballot boxes from an optimal solution that separate the plane into two parts such that each part contains between $k/3$ and $2k/3$ ballot boxes of the solution.

Related work The GERRYMANDERING problem was introduced by Lewenberg, Lev, and Rosenschein (2017). It is an example of general control problem, where a central agent may influence the outcome using its power over the voting process (Bartholdi, Tovey, and Trick 1992; Borodin et al. 2018; Faliszewski, Hemaspaandra, and Hemaspaandra 2010; Hemaspaandra, Hemaspaandra, and Rothe 2007; Lev and Lewenberg 2019). The problem also belongs to the large family of voting manipulation problems, whose computational aspects are widely studied in the computational social choice community, see (Conitzer and Walsh 2016; Faliszewski, Hemaspaandra, and Hemaspaandra 2014; Faliszewski and Procaccia 2010; Xia et al. 2009) for further references. In particular, parameterized complexity of manipulation was studied in (Dey, Misra, and Narahari 2015; 2016).

Preliminaries

For an integer i, we let $[i] = \{1, 2, \ldots, i\}$. We denote by $\mathbb{N}$ the set of natural numbers and by $\mathbb{R}$ the set of real numbers. Given two points $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$ in the plane $\mathbb{R}^2$, the *distance* between p_1 and p_2 is $\text{dist}(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

Voting and GERRYMANDERING

Given a set of *voters* $V = \{v_1, \ldots, v_n\}$ and a set of *candidates* C, the *preference ranking* of voter v_i is a total order (i.e., transitive, complete, reflexive, and antisymmetric relation) $\succ_i$ over C. The interpretation of $c_1 \succ_i c_2$ is that the voter v_i strictly prefers candidate c_1 over candidate c_2. We denote by $\pi(C)$ the set of preference rankings for the set of candidate C. A *voting rule* is then a function $f : \pi(C)^n \rightarrow C$. An *election* $\mathcal{E}_f = (C, V)$ is then comprised of a set of voters V, a set of candidates C, a preference ranking $\succ_i$ over C for each voter and a voting rule f. In this paper, we will only consider voting rules that do not distinguish between voters. That is the outcome of the voting rule only depends on the number of voters with each preference ranking and not their order. An example of such rule is *plurality*, in which for each voter only the topmost candidate counts and the winner is the candidate that got the most votes.

In a *district-based* election $\mathcal{E}_{f,g}$ voters are divided into disjoint sets $V_1, \ldots, V_s$ such that $\bigcup_{i \in [s]} V_i = V$. These sets define elections $\mathcal{E}^i_f = (C, V_i,)$. The ultimate winner from amongst the winners of $\mathcal{E}^i_f$ is determined by voting rule g, which for us will be a *threshold function*.

Definition 1 (GERRYMANDERING (Lewenberg, Lev, and Rosenschein 2017))**.** *The input of the* GERRYMANDERING$_f$ *problem is:*

- *A set of candidates C.*
- *A set of n voters $V = \{v_1, \ldots, v_n\} \subset \mathbb{R}^2$, where every voter $v_i \in V$ is identified by their location on the plane, and a preferred ranking $\succ_i \in \pi(C)$.*[1]
- *A set of m possible ballot boxes $B = \{b_1, \ldots, b_m\} \subset \mathbb{R}^2$.*
- *Parameters $\ell, k \in \mathbb{N}$, such that $\ell \leq k \leq m$.*
- *A target candidate $p \in C$.*

The task is to decide whether there is a subset of k ballot boxes $B' \subseteq B$, such that they define a district-based election, in which every voter votes at their closest ballot box in B', the winner at every ballot box is determined by voting rule f, and p wins in at least ℓ ballot boxes.

We assume that there are no two possible ballot boxes and a voter such that the voter is equidistant from the two boxes, meaning that no voter could ever be exactly on the boundary between two districts and so a fixed set of ballot boxes unambiguously defines the partition of voters into districts.

In Figure 1, we give an example of an input of GERRYMANDERING$_g$ and two possible solutions.

[1] Lewenberg et al. (Lewenberg, Lev, and Rosenschein 2017) also define the weighted problem, where every voter v has weight w_v. We focus on the version without weights.

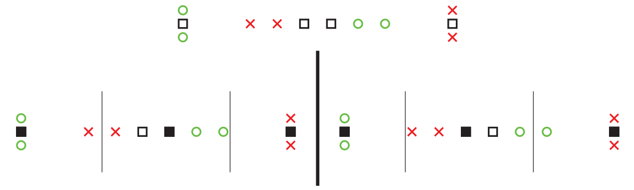

Figure 1: The example of (Lewenberg, Lev, and Rosenschein 2017). The green circle and red cross are voters voting for a different candidates. The squares are ballot boxes. In the bottom are two examples with 3 ballot boxes opened giving a different outcome.

Voronoi diagrams

To describe our algorithm, we require the notion of a Voronoi diagram. For a finite set of points P on the plane, *the Voronoi region* of a point $p \in P$ is the set of those points of the plane that are closer to p than to any other point in P. We may also use *the Voronoi cell* as a synonym. A Voronoi region is convex and its boundary is a polyline consisting of segments and possibly infinite rays, since the Voronoi region of $p \in P$ is essentially the intersection of $|P|-1$ half-planes. *The Voronoi diagram* of P is the tuple of the Voronoi regions of all the elements of P.

The Voronoi diagram could be considered a planar graph. The faces of the graph correspond to the Voronoi regions, the edges correspond to the segments which form the boundaries of the individual regions, and the vertices are the endpoints of these segments. The technical details of this correspondence can be found in (Marx and Pilipczuk 2015).

Voronoi diagrams appear naturally in GERRYMANDERING problem, since when the set S of ballot boxes chosen is fixed, the Voronoi diagram of S defines exactly which regions of the plane are assigned to vote in a particular ballot box.

Hardness

To obtain our ETH lower bound and prove W[1]-hardness of GERRYMANDERING, we reduce from the following W[1]-hard problem.

Definition 2 (GRID TILLING (Marx 2007))**.** *The input of the* GRID TILING *problem is:*

- *Integers $k, n \in \mathbb{N}$, and a collection $\mathcal{S}$ of k^2 nonempty sets $S_{i,j} \subseteq [n] \times [n] (1 \leq i, j \leq k)$.*

The task is to decide whether there is a set of pairs $s_{i,j} \in S_{i,j}$, for each $1 \leq i, j \leq k$, such that:

- *If $s_{i,j} = (p, q)$ and $s_{i+1,j} = (p', q')$, then $p = p'$.*
- *If $s_{i,j} = (p, q)$ and $s_{i,j+1} = (p', q')$, then $q = q'$.*

Lemma 1 (Marx 2007, see also Theorem 14.28 in Cygan et al. 2015)**.** GRID TILING *is* W[1]-hard *parameterized by k and, unless* ETH *fails, it has no $f(k)n^{o(k)}$-time algorithm for any computable function f.*

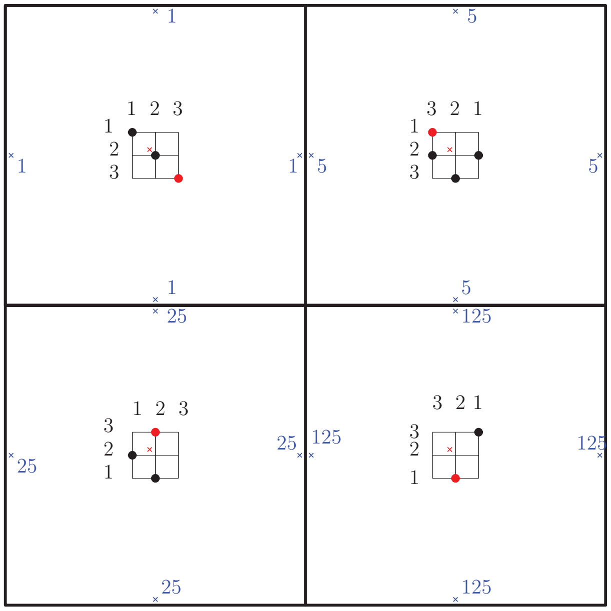

Figure 2: Example of a hardness reduction from Grid Tiling for $n = 3$ and $k = 2$. The sets are $S_{1,1} = \{(1,1),(2,2),(3,3)\}$, $S_{1,2} = \{(1,3),(2,1),(2,3),(3,2)\}$, $S_{2,1} = \{(1,2),(2,1),(3,2)\}$, and $S_{2,2} = \{(1,2),(3,1)\}$. The voters are denoted by ×, where our candidate is red and the other candidate is blue. The multiplicity of red voters in each cell is 1 more than the sum of the blue voters in the same cell. The ballot boxes are depicted as •. A solution to an instance is the set of red ballot boxes.

Overview of the Proof. Before we give a formal proof of Theorem 1, we first give an informal description of our reduction from GRID TILING problem that excludes some tedious computations that are necessary for the proof to work, but are not important for understanding where the difficulty of the problem lies. The main idea is to put all voters and ballot boxes inside large $k \times k$ grid such that the cell (i, j) of this grid correspond to the set $S_{i,j}$. In the center of each cell is one much smaller $n \times n$ grid corresponding to the set $S_{i,j}$ (see Figure 2). The ballot boxes are placed on appropriate positions inside small grid, with a small catch – for two neighboring cells of large grid, the small grids are "mirrored". This way, if we select precisely the same pair (p, q) for each set $S_{i,j}$, then the voters inside each large cell vote in the ballot box inside the same cell.

We will place a number of the "other candidate" voters, determined by a calculation later in the proof, very close to the center of each of 4 borders of the cell (see Figure 2). By selecting the right distances, for the size of the large grid, size of the small grid, and distance of the "other candidate" voters to the border of the cell, we can force that if we select ballot boxes corresponding to $s_{i,j} = (p, q) \in S_{i,j}$ and $s_{i+1,j} = (p', q') \in S_{i+1,j}$ (resp. $s_{i,j+1} = (p', q') \in S_{i,j+1}$), then the voters near the border between cells corresponding to $S_{i,j}$ and $S_{i+1,j}$ (resp. $S_{i,j+1}$) vote inside their corresponding cells if and only if $p = p'$ (resp. $q = q'$).

Finally, we place in the center of each large cell the number of the "our candidate" voters that is just 1 larger than the number of the "other candidate" voters inside the cell. As we placed "our candidate" voters on k^2 different positions, to win k^2 ballot boxes, in each box in a solution has to vote all voters from precisely one of these positions. Furthermore, by selecting the number of voters in each cell to grow exponentially, we get that the only way to win all ballot boxes is if all voters in the same cell vote in the same ballot box. This is only possible if we choose in each cell precisely one ballot box and if the corresponding selection of pairs in $S_{i,j}$'s is actually a solution to the original GRID TILING instance.

In the remaining of the section we formalize above intuition by assigning the voters and ballot boxes specific positions in the plane and computing distances between positions of ballot boxes and "other candidate" voters.

Theorem 1. *For any number* $|C| \geq 2$ *of candidates,* GERRYMANDERING$_{plurality}$ *is* W[1]-hard *parameterized by* $k+n$*. Moreover, there is no algorithm solving* GERRYMANDERING$_{plurality}$ *in time* $f(k,n) \cdot m^{o(\sqrt{k})}$ *for any computable function* f*, unless* ETH *fails.*

Proof. We will prove the theorem by a reduction from GRID TILING. Let $k, n \in \mathbb{N}$ be integers and $\mathcal{S}$ be a collection of k^2 nonempty sets $S_{i,j} \subseteq [n] \times [n] (1 \leq i,j \leq k)$. We will construct an instance $I = (C, V, B, k^2, k^2, \text{"red"})$ of GERRYMANDERING$_{plurality}$ with two candidates "red" and "blue", $2^{\mathcal{O}(k^2)}$ voters and $\mathcal{O}(k^2 n^2)$ ballot boxes and all voters and boxes positioned at the integral points inside $[\mathcal{O}(k \cdot n^2)] \times [\mathcal{O}(k \cdot n^2)]$ grid.

Ballot boxes. Let $(p,q) \in S_{i,j}$, we let $x^{i,j}_{(p,q)} = (10n^2 + 4n)(i-1) + 5n^2 + 4p$ if i is even and $x^{i,j}_{(p,q)} = (10n^2 + 4n)(i-1) + 5n^2 + 4(n-p)$ if i is odd. Similarly, we let $y^{i,j}_{(p,q)} = (10n^2 + 4n)(j-1) + 5n^2 + 4q$ if j is even and $y^{i,j}_{(p,q)} = (10n^2 + 4n)(j-1) + 5n^2 + 4(n-q)$ if j is odd. We then identify the pair $(p,q) \in S_{i,j}$ with a ballot box $b^{i,j}_{(p,q)} = (x^{i,j}_{(p,q)}, y^{i,j}_{(p,q)})$. Finally, we will denote by $B_{i,j}$ the set of ballot boxes associated with pairs in $S_{i,j}$.

Voters. For each pair of $1 \leq i,j \leq k$ we will place $5^{(i-1)\cdot k+(j-1)}$ "blue" voters at each of the following four positions: $((10n^2+4n)(i-1)+1, (10n^2+4n)(j-1)+5n^2+2n)$, $((10n^2+4n)(i-1)+5n^2+2n, (10n^2+4n)(j-1)+1)$, $((10n^2+4n)i-1, (10n^2+4n)(j-1)+5n^2+2n)$, and $((10n^2+4n)(i-1)+5n^2+2n, (10n^2+4n)j-1)$. We denote the sets of voters at the above four positions by $V^{i,j}_{\leftarrow}$, $V^{i,j}_{\uparrow}$, $V^{i,j}_{\rightarrow}$, and $V^{i,j}_{\downarrow}$, respectively. And we will place $4 \cdot 5^{(i-1)\cdot k+(j-1)} + 1$ "red" voters at the position $((10n^2+4n)+5n^2+2n, (10n^2+4n)(j-1)+5n^2+2n)$, we denote the set of these voters $V^{i,j}_{R}$.

The goal is to assign k^2 ballot boxes such that in each ballot box "red" has majority.

Correctness. We will show that $(k, n, \mathcal{S})$ is a YES-instance of GRID TILLING if and only if $(C, V, B, k^2, k^2, \text{"red"})$ is a YES-instance of GERRYMANDERING$_{plurality}$. Let $s_{i,j}$ be a pair in $S_{i,j}$ for all $i,j \in [k]$ and let $B' = \{b^{i,j}_{s_{i,j}} \mid i,j \in [k]\}$. The following two claims show that if $\bigcup_{i,j\in[k]}\{s_{i,j}\}$ is a solution to GRID TILLING, then B' is a solution to GERRYMANDERING$_{plurality}$. Moreover, if B' is a solution such that for all $i,j \in [k]$ the voters in $\bigcup_{\square\in\{R,\leftarrow,\rightarrow,\uparrow,\downarrow\}}$ vote in $b^{i,j}_{s_i,j}$, then $\bigcup_{i,j\in[k]}\{s_{i,j}\}$ is a solution to GRID TILLING.

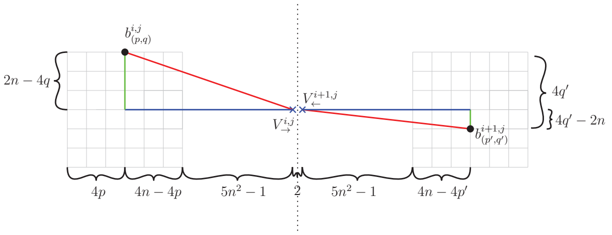

Figure 3: The situation in Claim 1. The distance $\text{dist}(b^{i,j}_{(p,q)}, V^{i,j}_{\rightarrow}) = \sqrt{(5n^2+4n-4p-1)^2+(2n-4q)^2}$. Which is clearly between $5n^2+4n-4p-1$ and $5n^2+4n-4p$. Similarly, $\text{dist}(b^{i+1,j}_{(p',q')}, V^{i+1,j}_{\rightarrow})$ is between $5n^2+4n-4p'-1$ and $5n^2+4n-4p'$.

Claim 1. *Let* $b^{i,j}_{(p,q)}$ *and* $b^{i+1,j}_{(p',q')}$ *be two ballot boxes. Then for every voter* $v \in V^{i,j}_{\rightarrow}$ *it holds* $\text{dist}(v, b^{i,j}_{(p,q)}) < \text{dist}(v, b^{i+1,j}_{(p',q')})$ *if and only if* $p \leq p'$*. Similarly, for every* $v' \in V^{i+1,j}_{\leftarrow}$ *it holds* $\text{dist}(v', b^{i+1,j}_{(p',q')}) < \text{dist}(v', b^{i,j}_{(p,q)})$ *if and only if* $p \geq p'$.

Proof. Let us assume that both i and j are even. The remaining 3 cases are analogous. Then $b^{i,j}_{(p,q)}$ is at position $((10n^2+4n)(i-1)+5n^2+4p, (10n^2+4n)(j-1)+5n^2+4q)$ and $b^{i+1,j}_{(p',q')}$ is at position $((10n^2+4n)i+5n^2+4(n-p'), (10n^2+4n)(j-1)+5n^2+4q')$. Moreover, voters in $V^{i,j}_{\rightarrow}$ and $V^{i+1,j}_{\leftarrow}$ are at positions $\big((10n^2+4n)i-1, (10n^2+4n)(j-1)+5n^2+2n\big)$ and $\big((10n^2+4n)i+1, (10n^2+4n)(j-1)+5n^2+2n\big)$, respectively.

Since both q and q' are numbers between 0 and n, it follows from Pythagorean theorem that the distance between $b^{i,j}_{(p,q)}$ and $V^{i,j}_{\rightarrow}$ is between $d^1_{\min} = 5n^2+4n-4p-1$ and $d^1_{\max} = \sqrt{(d^1_{\min})^2+(2n)^2} < d^1_{\min}+1$ (see Figure 3). For the same reasoning, the distance between $b^{i,j}_{(p,q)}$ and $V^{i+1,j}_{\leftarrow}$ is between $d^1_{\min}+2$ and $d^1_{\min}+3$, the distance between $b^{i+1,j}_{(p',q')}$ and $V^{i+1,j}_{\leftarrow}$ is between $d^2_{\min} = 5n^2+4n-4p'-1$ and $d^2_{\min}+1$, and finally the distance between $b^{i+1,j}_{(p',q')}$ and $V^{i,j}_{\rightarrow}$ is between $d^2_{\min}+2$ and $d^2_{\min}+3$. It follows that whenever $p > p'$, then $d^2_{\min}+3 < d^1_{\min}$ and the voters in $V^{i,j}_{\rightarrow}$ are closer to $b^{i+1,j}_{(p',q')}$ than to $b^{i,j}_{(p,q)}$. Similarly, if $p < p'$ then

$d^1_{\min} + 3 < d^2_{\min}$ and the voters in $V^{i+1,j}_{\leftarrow}$ are closer to $b^{i,j}_{(p,q)}$ than to $b^{i+1,j}_{(p',q')}$. □

Repeating the same argument for ballot boxes $b^{i,j}_{(p,q)}$ and $b^{i,j+1}_{(p',q')}$, we obtain also the following claim.

Claim 2. *Let $b^{i,j}_{(p,q)}$ and $b^{i,j+1}_{(p',q')}$ be two ballot boxes. Then for every voter $v \in V^{i,j}_{\downarrow}$ it holds $\mathrm{dist}(v, b^{i,j}_{(p,q)}) < \mathrm{dist}(v, b^{i,j+1}_{(p',q')})$ if and only if $q \leq q'$. Similarly, for every $v' \in V^{i,j+1}_{\uparrow}$ it holds $\mathrm{dist}(v', b^{i,j+1}_{(p',q')}) < \mathrm{dist}(v', b^{i,j}_{(p,q)})$ if and only if $q \geq q'$.*

From the above claims it directly follows that if we start with a YES-instance of GRID TILING, we end up with a YES-instance of GERRYMANDERING.

For the reverse direction, it remains to show that any solution B' to the reduced instance I of GERRYMANDERING has to select exactly one ballot box $b^{i,j}_{(p,q)}$ in each $B_{i,j}$ and that for each such box, the set of voters voting there is precisely the set of voters associated with (i, j).

First note that there are precisely k^2 positions of "red" voters. Hence each ballot box in a solution has to be closest to precisely one of these positions. Moreover, there are only k^2 more "red" voters than "blue" voters. Hence, if "red" wins in all boxes, then by pigeon-hole principle "red" has to win in each ballot box by precisely 1. Now let $b \in B'$ be a ballot box closest to $V^{i,j}_R$. Since $V^{i,j}_R$ is precisely the set of "red" voters that vote in b, none of voters in $\bigcup_{\square \in \{\leftarrow,\rightarrow,\uparrow,\downarrow\}} V^{i',j'}_{\square}$ such that $(i'-1)k + j' - 1 > (i-1)k + j - 1$ can vote in b. Now the number of the remaining "blue" voters is precisely $5 \cdot 5^{(i-1)k+(j-1)} - 1 = 4 \cdot \sum_{\ell=0}^{(i-1)k+(j-1)} 5^{\ell}$. Therefore, if one group of voters among $V^{i,j}_{\leftarrow}$, $V^{i,j}_{\uparrow}$, $V^{i,j}_{\rightarrow}$, and $V^{i,j}_{\downarrow}$ does not vote at b, then "red" wins b by more than 1 vote and by pigeon-hole principle, there will be a ballot box where "red" does not win. Finally, it is easy to see that if there is more than one ballot box selected from $B_{i,j}$, then the box where $V^{i,j}_R$ vote cannot take all four groups of voters $V^{i,j}_{\leftarrow}$, $V^{i,j}_{\uparrow}$, $V^{i,j}_{\rightarrow}$, and $V^{i,j}_{\downarrow}$. Therefore, it follows that in each $B_{i,j}$ exactly one ballot box is selected in B' and the set of voters in this ballot box is precisely the set of voters associated with (i, j). It follows from Claims 1 and 2 that the set of pairs associated with any such solution is a solution to GRID TILING. □

Algorithm

Our algorithm uses the machinery developed by Marx and Pilipczuk (Marx and Pilipczuk 2015) for guessing a balanced separator in the Voronoi diagram of a potential solution. The main idea of our algorithm is as follows. The Voronoi diagram of S could be viewed as a 3-regular planar graph with k faces and $\mathcal{O}(k)$ vertices. From (Marx and Pilipczuk 2015) we know that for such a graph there exists a polygon Γ which goes through $\mathcal{O}\left(\sqrt{k}\right)$ faces and vertices and there are at most $\frac{2}{3}k$ faces strictly inside Γ and at most $\frac{2}{3}k$ faces strictly outside Γ. Moreover, Γ could be defined by the sequence of $\mathcal{O}\left(\sqrt{k}\right)$ elements of B. This allows us to guess all possible variants of Γ without actually knowing the solution S, and for each Γ we recurse into two smaller subproblems defined inside and outside of Γ. Since the separator Γ is balanced and the recurrence $T(k) = n^{\mathcal{O}(\sqrt{k})} T(\frac{2}{3}k)$ solves to $T(k) = n^{\mathcal{O}(\sqrt{k})}$, we obtain the desired time bound. See Figure 4 for the illustration.

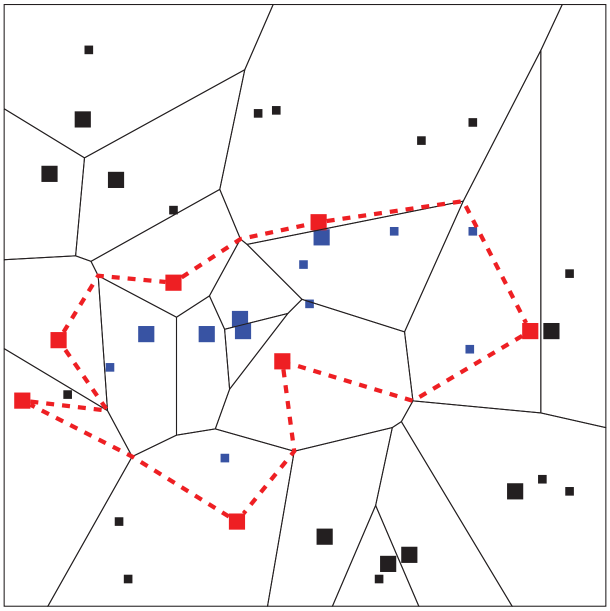

Figure 4: The Voronoi diagram of a potential solution. The solution is represented by large squares, small squares are the other possible ballot boxes. A separating polygon Γ is in red and dashed, the boxes on Γ are also in red. Our algorithm guesses Γ, takes the red boxes in the solution and then solves the inside (blue and red squares) and the outside (black and red squares) separately.

We assume that no four elements of B lie on the same circle. From this assumption, it follows that no vertex of any Voronoi diagram corresponding to a possible solution has degree more than 3. We also assume that for any four distinct points in B, one is never exactly the center of the unique circle passing through the three others. It means that for any subset S of B, the vertices of the Voronoi diagram of S are disjoint from B. Both assumptions are for the ease of presentation and could be achieved by a small perturbation of the coordinates.

Next we state the result of Marx and Pilipczuk about finding small balanced separators in Voronoi diagrams.

Lemma 2 (Marx and Pilipczuk 2015)**.** *Let S be a set of k points on the plane. Consider the Voronoi diagram of S, and the planar graph G associated with it. There is a polygon Γ which has length $\mathcal{O}\left(\sqrt{k}\right)$ and its vertices alternate between elements of S and vertices of G and each segment of Γ lies inside a face of G. At most $\frac{2}{3}k$ faces lie strictly inside Γ, and* *at most $\frac{2}{3}k$ strictly outside.*

Note that since Γ passes through $\mathcal{O}\left(\sqrt{k}\right)$ faces, the number of faces which lie non-strictly inside (outside) Γ is bounded by $\frac{3}{4}k$ when k is greater than γ, where γ is some constant.

Now we are ready to prove Theorem 2 which we restate for convenience.

Theorem 2. *For any constant number of candidates and for any voting rule g, GERRYMANDERING$_g$ is solvable in time $(m+n)^{\mathcal{O}(\sqrt{k})}$.*

Proof. Our algorithm is a recursive brute-force search. An input to the recursion step is a particular *state* $R = (V, B, \ell, k, F, v, \Delta)$. A state consists of a set of voters V, a set of possible ballot boxes B, parameters ℓ, k such that $\ell \le k \le |B|$, a subset of boxes $F \subset B$, for any $f \in F$ and $\sigma \in \pi(C)$ there is an integer $v(f, \sigma)$ from 0 to $|V|$, and a set of segments Δ, the segments form the boundary of the region containing V and B. Each segment $\delta \in \Delta$ has exactly one endpoint in F, and we denote this endpoint by δ_F. The algorithm returns whether the given state is *valid*.

Definition 3 (Valid state)**.** *A state is valid if there exists a subset $S \subset B \setminus F$ such that $|S|+|F| = k$, and in the election with voters V and boxes $S \cup F$ the target party p wins in at least ℓ boxes of S, and for each $f \in F$ and $\sigma \in \pi(C)$ there are exactly $v(f, \sigma)$ voters with preference list σ voting in box f. Additionally, in the Voronoi diagram of $S \cup F$, each segment δ of Δ lies completely inside the cell (touching the border) corresponding to the box δ_F.*

If the state is valid, the algorithm also returns a particular subset $S \subset B \setminus F$ satisfying Definition 3.

Intuitively, V and B correspond to a set of voters and possible boxes in the current subtask, F corresponds to a set of boxes which we have already decided to take into the solution on the previous steps of the recursion, and for any box in F the number of voters with a particular preference list is also fixed through v. Among the boxes of $B \setminus F$ we are supposed to select at most $k - |F|$ to go into the solution, and in at least ℓ of them the target party should win. The condition that segments of Δ lie inside the corresponding cells enforces that these cells actually separate the whole state from the outside of the region defined by Δ.

Initially, to run our algorithm having the input (C, V, B, ℓ, k, p) to GERRYMANDERING$_f$, we proceed as follows. First, find an equilateral triangle T such that V and B are completely inside T. Then mirror each vertex of the triangle against the opposing side, and denote the resulting set of three points as F. Consider the Voronoi diagram of $B \cup F$, by construction the outer faces are exactly the cells of F, and these faces are disjoint from B and V. Construct a polygon Δ in the following way. Start from a point of F and go to a next clockwise point of F by a sequence of two segments having as the common point a vertex of the diagram on the border between the corresponding outer faces; repeat until the polygon reaches the starting point again.The initial state is defined as $R_0 = (V, B \cup F, \ell, k+3, F, v, \Delta)$ where $v(f, \sigma) = 0$ for any $f \in F$ and $\sigma \in \pi(C)$. Since for any $v \in V$ any $b \in B$ is closer to v than any $f \in F$, R_0 is valid if and only if (C, V, B, ℓ, k, p) is a YES-instance of GERRYMANDERING$_f$.

The recursive step. On the given state $R = (V, B, \ell, k, F, v, \Delta)$ the algorithm proceeds as follows. If $k - |F|$ is at most γ, where γ is the constant mentioned below the statement of Lemma 2, we try all possible $S \subset B$ of size $k - |F|$ and for each check the conditions of validity in polynomial time, since $\gamma = O(1)$ the whole procedure works in polynomial time as well. If $k - |F| > \gamma$, try all possible polygons Γ of the form as in Lemma 2, that is, alternating between elements of B and potential vertices of the Voronoi diagram of the solution, and having length at most $\alpha\sqrt{k}$ where α is the constant under $\mathcal{O}$ in the lemma. Since a vertex of a Voronoi diagram constructed over any subset of B is equidistant from three elements from this subset and is uniquely determined by these three elements, there are at most $|B|^3$ potential locations for a Voronoi diagram vertex. Therefore there are at most $|B|^{4\alpha\sqrt{k}}$ variants for Γ. We consider only these Γ which do not go out of the region defined by Δ.

For each possible Γ, we split the instance in two parts. Denote by Q the set of boxes on Γ. Let V_1 (V_2) be the set of voters inside (outside) of Γ, and B_1 (B_2) be the set of possible boxes inside (outside) of Γ, $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$, $B_1 \cup B_2 = B$. For $i \in \{1, 2\}$, let $F_i = Q \cup (F \cap B_i)$ and let Δ_i be those segments δ in Δ such that $\delta_F \in F_i$. The fixed preference counts v_1 and v_2 for F_1 and F_2 respectively are defined as follows. For boxes in $F \setminus Q$ the value of v is transferred directly to v_1 or v_2 depending on to which part the box went. For boxes in $F \cap Q$ and for any preference list $\sigma \in \pi(C)$ we guess how the value of $v(f, \sigma)$ is split between $v_1(f, \sigma)$ and $v_2(f, \sigma)$. For boxes in $Q \setminus F$ we guess separately the value of $v_1(f, \sigma)$ and $v_2(f, \sigma)$.

Finally, guess how $k + |Q \setminus F|$ is split between the two parts k_1 and k_2, k_1 and k_2 must be each at most $\frac{3}{4}k$. Also guess how $\ell - w$ is split between ℓ_1 and ℓ_2, where w is the number of boxes in $Q \setminus F$ where the target party p wins if voters from V_1 and V_2 vote according to v_1 and v_2.

Next, we run the recursion on the state $R_1 = (V_1, B_1, \ell_1, k_1, F_1, v_1, \Delta_1 \cup \Gamma)$ and on the state $R_2 = (V_2, B_2, \ell_2, k_2, F_2, v_2, \Delta_2 \cup \Gamma)$. If both R_1 and R_2 are reported as valid states, we return that R is valid and take $S = S_1 \cup S_2 \cup (Q \setminus F)$. Otherwise, we continue to the next choice of Γ. If for all choices of Γ we do not succeed, we return that R is not valid.

The correctness. Now we prove the correctness of the algorithm above. The proof is by induction on k. In the base case $k - |F| \le \gamma$ the algorithm tries all possible ways to select the ballot boxes and then checks the definition directly. In the case $k - |F| > \gamma$, assume first that the algorithm reports a given state $R = (V, B, \ell, k, F, v, \Delta)$ as valid. So for some Γ the corresponding splitting states $R_1 = (V_1, B_1, \ell_1, k_1, F_1, v_1, \Delta_1 \cup \Gamma)$ and $R_2 = (V_2, B_2, \ell_2, k_2, F_2, v_2, \Delta_2 \cup \Gamma)$ are reported as valid by the algorithm, and by induction that means that they are actually valid states since both k_1 and k_2 are at most $\frac{3}{4}k$.

Consider sets $S_1 \subset B_1 \setminus F_1$ and $S_2 \subset B_2 \setminus F_2$ returned by the recursive calls, by induction S_1 and S_2 also satisfy Definition 3 on the validity of R_1 and R_2, respectively. As before, denote $S_1 \cup S_2 \cup (Q \setminus F)$ by S. We claim that S satisfies Definition 3 for R, and therefore R is valid.

By construction of R_1 and R_2, $k = k_1 + k_2 - |Q \setminus F|$, and since S_1, S_2, and $Q \setminus F$ are pairwise disjoint, the size of S is indeed equal to $k - |F|$. Next, we show that in the election with boxes $S \cup F$ the target party p wins in at least ℓ districts of S and that the votes in districts of F are distributed according to v. The next claim shows that for voters in each part the box where they vote is the same in the election with boxes $S_i \cup F_i$ and in the election with boxes $S \cup F$.

Claim 3. *For $i \in \{1, 2\}$ and any voter $x \in V_i$, the closest box to x in $S \cup F$ belongs to $S_i \cup F_i$ as well.*

Proof. Let $i = 1$, the other case is analogous. Assume the contrary, there is a voter $x \in V_1$ such that there is a box $b \in S_2 \cup F_2 \setminus Q$ which is strictly closer to x than any box in $S_1 \cup F_1$, since two boxes cannot be at the same distance to x. Consider a straight line segment from x to b, since x and b lie on the different sides of Γ, the segment crosses Γ at least once, denote any intersection point by y. Since R_2 is valid, the segments of Γ lie completely inside the cells of Q in the Voronoi diagram of $S_2 \cup F_2$. That means that there is $q \in Q$ such that y is at least as close to q as to any other box in $S_2 \cup F_2$, including b. But then $\operatorname{dist}(x, q) \leq \operatorname{dist}(x, y) + \operatorname{dist}(y, q) \leq \operatorname{dist}(x, y) + \operatorname{dist}(y, b) = \operatorname{dist}(x, b)$, where the first inequality is the triangle inequality, and this contradicts the initial assumption. □

By Claim 3, for each $i \in \{1, 2\}$ and for each box in S_i exactly the same set of voters votes there in the election with boxes $S \cup F$ compared to the election with boxes $S_i \cup F_i$. It means that among the boxes in S_1 and S_2 the target party p wins in exactly $\ell_1 + \ell_2$ of them. For each $i \in \{1, 2\}$ and each box f in $F_i \setminus Q$ it also holds that exactly the same set of voters votes in f in the election with boxes $S \cup F$ as votes in f in the election with boxes $S_i \cup F_i$. Since $v(f, \sigma) = v_i(f, \sigma)$ for any $\sigma \in \pi(C)$, the vote distribution on $F_i \setminus Q$ is as desired. For each box $q \in Q$ and each $\sigma \in \pi(C)$ by construction $v(q, \sigma) = v_1(q, \sigma) + v_2(q, \sigma)$. Since voters from V_1 and V_2 who vote in q are exactly preserved, $v(q, \sigma)$ is indeed equal to the number of voters with preference list σ who vote in q. This finishes the proof that R is valid and S satisfies Definition 3.

Finally, we show that for each $\delta \in \Delta$, δ lies inside the cell of δ_F in the Voronoi diagram of $S \cup F$. If $\delta_F \in Q$, since R_1 and R_2 are valid, δ lies inside the cell of δ_F in the Voronoi diagram of $S_i \cup F_i$ for $i \in \{1, 2\}$, then no other point in $S \cup F$ is closer to each point on δ than δ_F. If $\delta_F \notin Q$, δ is completely inside or outside of Γ, and the same argument as in Claim 3 shows that for any point on δ the closest box is preserved, then it has to be δ_F, since δ is in Δ_i for some $i \in \{1, 2\}$, and R_i is valid.

Towards the other direction, assume that R is valid, we show that the algorithm correctly reports the validity of R. Consider a particular S from Definition 3, by Lemma 2 for the Voronoi diagram of $S \cup F$, there exists a polygon Γ of length $O(\sqrt{k})$ which is a balanced separator for $S \cup F$. Since the algorithm tries all polygons of this form, it will try Γ as well, or report that R is valid earlier. When considering the polygon Γ, the algorithm will guess the right values of k_1, k_2, ℓ_1, ℓ_2, and v_1, v_2 on the boxes of $Q \setminus F$, according to the elections on $S \cup F$.The new states R_1 and R_2 are valid since the elections on $S \cap F$ exactly induce conditions on R_i, and there exists a valid selection of boxes $S_i = (S \cup F) \cap B_i \setminus Q$, for each $i \in \{1, 2\}$. Therefore the recursive call will find R_i valid by induction. Thus the proof of correctness is finished.

Running time analysis. Denote by $T(k)$ the worst-case running time of our algorithm where k is the respective parameter of the input state. If $k \leq \gamma$, the algorithm does a brute force in polynomial time so $T(k) = (n + m)^{\mathcal{O}(1)}$. If $k > \gamma$, we try at most $m^{4\alpha\sqrt{k}}$ polygons Γ, for each of them we in time at most $k \ell n^{2\alpha\sqrt{k}}$ guess the parameters of the new two instances k_1, ℓ_1, v_1 and k_2, ℓ_2, v_2. We run our algorithm on the two instances recursively, and in both instances the value of the parameter is bounded by $\frac{3}{4}k$, so we get the following recurrence relation,

$$\begin{cases} T(k) \leq (n+m)^{\beta\sqrt{k}} T(\frac{3}{4}k), & k > \gamma \\ T(k) \leq (n+m)^{\beta}, & k \leq \gamma, \end{cases}$$

where β is some constant. Now, by applying the first upper bound until the expression under T is at most γ we get

$$\begin{aligned} T(k) &\leq (n+m)^{\beta\sqrt{k}(1+q+q^2+\ldots+q^t)} \\ &\leq (n+m)^{\beta\sqrt{k}/(1-q)} = (n+m)^{\mathcal{O}(\sqrt{k})}, \end{aligned}$$

where $q = \sqrt{3/4}$ and $t = \lceil \log_{4/3}(k/\gamma) \rceil$. □

Concluding Remarks

We initiated study of a gerrymandering problem from the perspective of fine-grained and parameterized complexity. Our main result is asymptotically tight algorithm with respect to parameter k, the number of opened ballot boxes, together with a matching lower bound.

For future work, it would be interesting to investigate the complexity of finding an approximate solution for the problem. That is the question whether there is an efficient (polynomial time) algorithm that either finds a set of k ballot boxes such that p wins $\frac{\ell}{c}$ districts, for some constant c, or correctly decides that in no elections with k districts the candidate p wins ℓ of them. Another interesting question is what happens, when we introduce obstacles, such as lakes or hills, in the instance so that the distances are not anymore Euclidean distances in the plane. Can we still get better than trivial m^k algorithm in this case? Finally, it is also natural to impose restrictions on sizes of districts, so they are not too disproportional.

Acknowledgments

This work is supported by the Research Council of Norway via the project “MULTIVAL”.

References

Bartholdi, III, J. J.; Tovey, C. A.; and Trick, M. A. 1992. How hard is it to control an election? *Math. Comput. Model.* 16(8-9):27–40.

Borodin, A.; Lev, O.; Shah, N.; and Strangway, T. 2018. Big city vs. the great outdoors: Voter distribution and how it affects gerrymandering. In *Proc. IJCAI 2018*, 98–104. IJCAI/AAAI Press.

Conitzer, V., and Walsh, T. 2016. Barriers to manipulation in voting. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*. Cambridge University Press. 127–145.

Cygan, M.; Fomin, F. V.; Kowalik, L.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized Algorithms*. Springer.

Dey, P.; Misra, N.; and Narahari, Y. 2015. Kernelization complexity of possible winner and coalitional manipulation problems in voting. In *Proc. AAMAS 2015*, 87–96. ACM.

Dey, P.; Misra, N.; and Narahari, Y. 2016. Complexity of manipulation with partial information in voting. In *Proc. IJCAI 2016*, 229–235. IJCAI/AAAI Press.

Downey, R., and Fellows, M. 1999. *Parameterized Complexity*. New York: Springer.

Downey, R. G., and Fellows, M. R. 2013. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer.

Faliszewski, P., and Procaccia, A. D. 2010. AI's war on manipulation: Are we winning? *AI Magazine* 31(4):53–64.

Faliszewski, P.; Hemaspaandra, E.; and Hemaspaandra, L. A. 2010. Using complexity to protect elections. *Commun. ACM* 53(11):74–82.

Faliszewski, P.; Hemaspaandra, E.; and Hemaspaandra, L. A. 2014. The complexity of manipulative attacks in nearly single-peaked electorates. *Artificial Intelligence* 207:69–99.

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

Impagliazzo, R., and Paturi, R. 2001. On the complexity of k-SAT. *Journal of Computer and System Sciences* 62(2):367–375.

Lev, O., and Lewenberg, Y. 2019. "Reverse Gerrymandering": a decentralized model for multi-group decision making. In *Proc. AAAI 2019*. AAAI Press.

Lewenberg, Y.; Lev, O.; and Rosenschein, J. S. 2017. Divide and conquer: Using geographic manipulation to win district-based elections. In *Proc. AAMAS 2017*, 624–632. ACM.

Lokshtanov, D.; Marx, D.; and Saurabh, S. 2011. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS* 105:41–72.

Marx, D., and Pilipczuk, M. 2015. Optimal Parameterized Algorithms for Planar Facility Location Problems Using Voronoi Diagrams. In *Proc. ESA 2015*, 865–877. Berlin, Heidelberg: Springer Berlin Heidelberg.

Marx, D. 2007. On the optimality of planar and geometric approximation schemes. In *Proc. FOCS 2007*, 338–348. IEEE.

Terbush, J. 2013. How to fix gerrymandering and stop future shutdowns.

Xia, L.; Zuckerman, M.; Procaccia, A. D.; Conitzer, V.; and Rosenschein, J. S. 2009. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proc. IJCAI 2015*, 348–353. AAAI Press.