

Reconfiguration of Connected Graph Partitions*

Hugo A. Akitaya[†] Matthew D. Jones[‡] Matias Korman[§] Oliver Kortén[¶]
Christopher Meierfrankenfeld^{||} Michael J. Munje^{**} Diane L. Souvaine^{||}
Michael Thramann^{||} Csaba D. Tóth^{††}

June 30, 2021

Abstract

Motivated by recent computational models for redistricting and detection of gerrymandering, we study the following problem on graph partitions. Given a graph G and an integer $k \geq 1$, a **k -district map** of G is a partition of $V(G)$ into k nonempty subsets, called **districts**, each of which induces a connected subgraph of G . A **switch** is an operation that modifies a k -district map by reassigning a subset of vertices from one district to an adjacent district; a **1-switch** is a switch that moves a single vertex. We study the connectivity of the configuration space of all k -district maps of a graph G under 1-switch operations. We give a combinatorial characterization for the connectedness of this space that can be tested efficiently. We prove that it is PSPACE-complete to decide whether there exists a sequence of 1-switches that takes a given k -district map into another; and NP-hard to find the shortest such sequence (even if a sequence of polynomial length is known to exist). We also present efficient algorithms for computing a sequence of 1-switches that takes a given k -district map into another when the space is connected, and show that these algorithms perform a worst-case optimal number of switches up to constant factors.

1 Introduction

An **electoral district** is a subdivision of territory used in the election of members to a legislative body. **Gerrymandering** is the practice of drawing district boundaries with the intent to give political advantage to a particular group; it tends to occur in electoral systems that elect one representative per district. Detecting whether gerrymandering has been employed in designing a given district map and producing unbiased district maps are important problems to ensure fairness in the outcome of elections. Numerous quality measures have been proposed for the comparison of district maps [10, 11], but none of them is known to eliminate bias. Research has focused on exploring the space of all possible district maps that meet certain basic criteria. Since this space is computationally intractable, even for relatively small instances, randomized algorithms play an important role in finding “average” district maps under suitable distributions [3]. Being an outlier may indicate that gerrymandering has been applied in the drawing of a given map [21].

*Research supported in part by NSF CCF-1422311, CCF-1423615, and OIA-1937095.

[†]Department of Computer Science, University of Massachusetts Lowell, Lowell, MA, USA.

[‡]Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA.

[§]Siemens Electronic Design Automation, Wilsonville, OR, USA.

[¶]Department of Computer Science, Columbia University, New York, NY, USA.

^{||}Department of Computer Science, Tufts University, Medford, MA, USA.

^{**}Department of Computer Science, California State University Northridge, Los Angeles, CA, USA.

^{††}Department of Mathematics, California State University Northridge, Los Angeles, CA, USA.

Fifield et al. [14] model a district map as a vertex partition on an adjacency graph of census tracts or voting precincts. A **census tract** is a small territorial subdivision used as a geographic unit in a census. Each district corresponds to a set of census tracts in the partition and must induce a connected subgraph. The graphs currently used in practice are the dual graphs of a terrain partition, where two districts are adjacent if and only if their boundaries intersect in at least one point. Because of degeneracies, five “wedge-like” districts may meet at a single point and induce a K_5 in the dual graph.¹ In particular, the district maps are not necessarily planar.

Starting from a given district map, one can obtain another map by switching a subset of census tracts from one district to another. The goal is to apply a sequence of such operations randomly, and arrive at a uniformly random sample of the space of all possible district maps that meet the desired criteria. Under some assumptions, Fifield et al. [14] proves that the Markov chain produced by their experiments is ergodic². More interestingly, if the assumptions hold, it will have a unique stationary distribution, which is approximately uniform on the space of all k -district maps. One of the assumptions is that the underlying sample space is connected under the switch operation. However, connectedness is only assumed and remains unproven in [14].

In this paper, we provide a rigorous graph-theoretic background for studying the space of district maps with a given number of districts. We focus on the 1-switch operation that moves precisely one vertex from one district to an adjacent one. The remainder of the paper will call such an operation simply a “switch.” Other than requiring connectedness of districts, we do not impose any other restrictions on the district maps. In particular, the size of a district can be any integer in the range $[1, n - k + 1]$ where n is the number of census tracts.

The fact that the space of all k -district maps is connected in our model implies that any aperiodic Markov chain is also ergodic on the subset of k -district maps that meet the desired criteria. Thus, our results have implications for models with additional desired criteria (other than connectivity). A natural criterion relevant for the gerrymandering setting is that district maps remain **balanced** (that is, all sets have roughly the same size). Besides being an important step in showing theoretical soundness of a Markov-chain-based sampling approach, our results demonstrate how the connectivity of the space relates to how well the adjacency graph is connected. This in turn helps design new operations to traverse the space, and provides a framework for comparing them.

Our Results. We consider the graph-theoretic model from [14]. For an n -vertex graph G (the adjacency graph of precincts or census tracts) and an integer $1 \leq k \leq n$, we consider the **switch graph** $\Gamma_k(G)$ in which each node corresponds to a partition of $V(G)$ into k nonempty subsets (districts), each of which induces a connected subgraph of G , and an edge corresponds to switching one vertex from one district to an adjacent district (see Section 2 for a definition). We do not assume planarity of G unless noted otherwise.

1. **Connectedness.** We prove that $\Gamma_k(G)$ is connected if G is biconnected (Theorem 5), and give a combinatorial characterization of the connectedness of $\Gamma_k(G)$ that can be tested in $O(n + m)$ time, where $n = |V(G)|$ and $m = |E(G)|$ (Theorem 17). In general, however, it is PSPACE-complete to decide whether two given nodes of $\Gamma_k(G)$ are in the same connected component even when G is planar (Theorem 22), or G is nonplanar and $k = 2$ (Theorem 23).
2. **Shrinkable Districts.** One of our key methods to modify a district map is to shrink a district into a single vertex by a sequence of switch operations. If this is feasible, we call the district **shrinkable**; if all districts are shrinkable, we call the district map **shrinkable**. We

¹Similar phenomenon occurs around a lake, where all districts adjacent to the water are pairwise adjacent.

²A Markov chain is *ergodic* if it is aperiodic and positive recurrent (that is, each state has a positive probability to be revisited, see [14] for more details).

prove that the subgraph $\Gamma'_k(G)$ of $\Gamma_k(G)$ induced by shrinkable district maps is connected if G is connected (Theorem 9).

3. **Diameter.** When G is biconnected, the diameter of $\Gamma_k(G)$ is in $O(kn)$, where $n = |V(G)|$ (Theorem 9), and this bound is the best possible (Theorem 8). When $\Gamma_k(G)$ is disconnected, the diameter of a component may be as large as $2^{\Omega(n)}$ even for planar graphs (Corollary 25).
4. **Shortest Path.** Finding the distance between two nodes of $\Gamma_k(G)$ is NP-hard, even if $\Gamma_k(G)$ is connected (Theorem 29).

Related Previous Work. Graph partitions and graph clustering algorithms [36] are widely used in divide-and-conquer strategies. These algorithms, however, do not explore the space of all partitions into k connected subgraphs. Evolutionary algorithms [35], in this context, modify a partition by random “mutations,” which are successive coarsening and uncoarsening operations, rather than moving vertices from one subgraph to another.

While the adjacency graph model for district maps has been used for decades in combinatorial optimization and operations research [33], the objective was finding optimal district maps under one or more criteria. Since exhaustive search is infeasible and most variants of the optimization problem are intractable [30], local search heuristics were suggested [34]. Several combinatorial results restrict G to be a square grid [1, 29]. Heuristic and intractability results are also available for geometric variants of the optimization problem, where districts are polygons in the plane [15, 25, 32].

Elementary graph operations similar to our 1-switch operation have also been studied. Motivated by the classical “Fifteen” puzzle, Wilson [37] studied the configuration space of t , $t < n$, indistinguishable pebbles (a.k.a. tokens [28]) on the vertices of a graph G with n vertices, where each pebble occupies a unique vertex of G , and can move to any adjacent unoccupied vertex. The occupied and unoccupied vertices partition $V(G)$ into two subsets. Crucially, the number of pebbles is fixed, and the occupied vertices need not induce a connected subgraph. Results include a combinatorial characterization of the configuration space (a.k.a. token graph) [37], NP-hardness for deciding connectedness [24], finding the shortest path between two configurations [18, 31], and bounds on the diameter and the connectivity of the configuration space [26, 28]. Demaine et al. [8] considered a subgraph of the token graph, where the tokens are located at an independent set. The diameter and shortest path in the configuration space can often be computed efficiently when the underlying graph G is a tree [2, 8], or chordal [5]. There are a few results that require the occupied vertices to induce a connected subgraph, but they are limited to the case where G is a grid [12, 23], and the number of pebbles is still fixed.

Goraly et al. [19] later considered colored pebbles (tokens). Each color class consists of indistinguishable pebbles, unoccupied vertices are considered as one of the color classes [17, 38, 39]: Hence all vertices in $V(G)$ are occupied and a move can swap the pebbles on two adjacent vertices. The color classes (including the “unoccupied” color) partition $V(G)$ into subsets. However, the cardinality of each color class remains fixed and the color classes need not induce connected subgraphs. Results, again, include combinatorial characterizations to connected configuration space [16], NP-completeness for the connectedness of the configuration space for $k \geq 3$ colors, and a polynomial-time algorithm for finding the shortest path for $k = 2$ colors. See [6, 27] for recent results on the parametric complexity of these problems.

The problem of partitioning a graph G into k connected subgraphs with equal (or almost equal) number of vertices is known as the **Balanced Connected k -Partition Problem** (BCP_k), which is NP-hard already for $k = 2$ [13], for grids in general [4], and also hard to approximate within an absolute error of $n^{1-\delta}$ [7].

Organization. Section 2 defines the reconfiguration problem formally, and describes some important properties of shrinkable districts. Section 3 shows that $\Gamma_k(G)$ is connected if G is biconnected, and $\Gamma'_k(G)$ is connected if G is connected. Section 4 presents our PSPACE-completeness proof and lower bounds for the diameter of $\Gamma_k(G)$, and Section 5 continues with our NP-hardness results for the shortest path problem. We conclude in Section 6 with open problems.

2 Preliminaries

Let $G = (V, E)$ be a connected graph. A k -**district map** Π of G is a partition of $V(G)$ into disjoint nonempty subsets $\{V_1, \dots, V_k\}$ such that the subgraph induced by V_i is connected for all $i \in \{1, \dots, k\}$. Each subgraph induced by V_i is called a **district**. We abuse the notation by writing $\Pi(v)$ for the subset in Π that contains vertex v . We now formally define the switch operation. Our definition matches the previous informal description. Given a k -district map $\Pi = \{V_1, \dots, V_k\}$, and a path (u, v, w) in G such that $\Pi(u) = \Pi(v) \neq \Pi(w)$, a **switch** (denoted $\text{switch}_\Pi(u, v, w)$) is an operation that returns a k -district map obtained from Π by removing v from the subset $\Pi(u)$ and adding it to $\Pi(w)$. More formally,

$$\text{switch}_\Pi(u, v, w) = \Pi' = (\Pi \setminus \{\Pi(u), \Pi(w)\}) \cup \{\Pi(u) \setminus \{v\}, \Pi(w) \cup \{v\}\}$$

if Π' is a k -district map. Note that $\text{switch}_\Pi(u, v, w)$ is not defined if $\Pi(v) \setminus \{v\}$ induces a disconnected subgraph. A switch is always reversible since if $\text{switch}_\Pi(u, v, w) = \Pi'$, then $\text{switch}_{\Pi'}(w, v, u) = \Pi$. We may omit the subscript when the map in which the switch is applied is clear from context. For every graph G and integer k , the **switch graph** $\Gamma_k(G)$ is the graph whose vertex set is the set of all k -district maps of G , and $\Pi_1, \Pi_2 \in V(\Gamma_k(G))$ are connected by an edge if there exist $u, v, w \in V(G)$ such that $\text{switch}_{\Pi_1}(u, v, w) = \Pi_2$.

2.1 Block Trees and SPQR Trees

Biconnectivity plays an important role in our proofs. In particular, we rely on the concept of a **block tree**, which represents the containment relation between the blocks (maximal biconnected components) and the cut vertices of a connected graph, and a **SPQR tree**, which is a recursive decomposition of a biconnected graph. We review both concepts here.

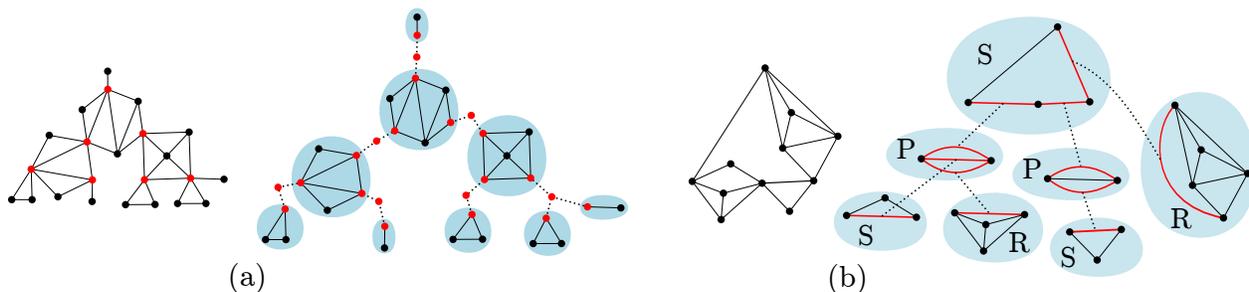


Figure 1: (a) A connected graph and its block tree. Cut vertices are shown in red, and dotted lines connect two occurrences of the same vertex in adjacent nodes of the block tree. (b) A 2-connected graph and its SPQR tree. Virtual edges are shown in red. Corresponding pairs of virtual edges are connected with dotted lines.

Block Trees. Let G be a connected graph. Let $B(G)$ be the set of **blocks** of G . (Two adjacent vertices induce a 2-connected subgraph, so a block may be a subgraph with a single edge.) Let $C(G)$

be the set of cut vertices in G . Then the **block tree** $T = T(G)$ is a bipartite graph, whose vertex set is $V(T) = B(G) \cup C(G)$, and T contains an edge $(W, c) \in B(G) \times C(G)$ if and only if $c \in W$ (i.e., block W contains vertex c). The definition immediately implies that a leaf and its unique neighbor induce a block $W \in B(G)$ (and never a cut vertex in $C(G)$). The block tree can be computed in $O(|E(G)|)$ time and space [22]. For convenience, we label every biconnected component by its vertex set (i.e., for a block $W \in B(G)$, we denote by W the set of vertices in the block).

SPQR Trees. Let G be a biconnected graph. A deletion of a (vertex) 2-cut $\{u, v\}$ disconnects G into two or more components $C_1, \dots, C_i, i \geq 2$. A **split component** of $\{u, v\}$ is the subgraphs of G induced by $V(C_j) \cup \{u, v\}$ for $j = 1, \dots, i$, or the graph induced by $\{u, v\}$ if $uv \in E(G)$. The SPQR-tree T_G of G represents a recursive decomposition of G defined by its 2-cuts. A node μ of T_G is associated with a multigraph called **skeleton**(μ) on a subset of $V(G)$ obtained by adding a **virtual** edge uv to a split component of the 2-cut $\{u, v\}$, or by creating a virtual (parallel) edge uv for each split component of $\{u, v\}$. Hence, an edge in **skeleton**(μ) is **real** if it is an edge in G , or virtual otherwise. A node μ has a **type** in $\{S, P, R\}$. If the type of μ is S, then **skeleton**(μ) is a cycle of 3 or more vertices. If the type of μ is P, then **skeleton**(μ) consists of 3 or more parallel edges between a pair of vertices. If the type of μ is R, then **skeleton**(μ) is a 3-connected graph on 4 or more vertices. Two nodes μ_1 and μ_2 of T_G are adjacent if **skeleton**(μ_1) and **skeleton**(μ_2) share exactly two vertices, u and v , that form a 2-cut in G . Each virtual edge in **skeleton**(μ) has a corresponding pair in **skeleton**(μ') for some adjacent node μ' ; see Figure 1(b). The graph G can be reconstructed from the skeletons of the nodes in T_G by identifying every pair of corresponding virtual edges and then deleting all virtual edges. No two S nodes (resp., no two P nodes) are adjacent. Therefore, T_G is uniquely defined by G . If μ is a leaf in T_G , then **skeleton**(μ) has a unique virtual edge; in particular the type of every leaf is S or R. The SPQR tree T_G has $O(|E(G)|)$ nodes and can be computed in $O(|E(G)|)$ time [9].

2.2 Shrinkability

Consider a graph G and a k -district map Π . We say that the operation $\text{switch}_\Pi(u, v, w)$ **shrinks** $\Pi(u)$ to $\Pi(u) \setminus \{v\}$, and **expands** $\Pi(w)$ to $\Pi(w) \cup \{v\}$. A sequence of switches **shrinks** (resp., **expands**) V_i to V'_i if there exists a sequence of consecutive switches that jointly shrink (resp., expand) V_i to V'_i . A subset $V_i \in \Pi$ (and its induced district) is **shrinkable** if it can be shrunk to a singleton (district of size one) by a sequence of $|V_i| - 1$ switches; otherwise it is **unshrinkable**. A k -district map is **shrinkable** if each of its districts is shrinkable. A district V_i is said to **contain** a block $W \in B(G)$ if it contains all vertices in W .

In the remainder of this section we state some simple properties that will be used later.

Lemma 1. *A switch operation cannot move a leaf of G from one district to another.*

Proof. Let $v \in V(G)$ be a leaf in G , and let $u \in V(G)$ be its unique neighbor. Since v is a leaf there is no path (u, v, w) and hence there is no valid $\text{switch}_\Pi(u, v, w)$ moving v to another district. \square

Lemma 2. *Let T be the block tree of a graph G , and let Π be a k -district map on G . If a district V_ℓ contains two leaves of T , say $W_i, W_j \in B(G)$, then a switch operation cannot move any vertex from $W_i \cup W_j$ to another district. Consequently, V_ℓ is unshrinkable.*

Proof. Suppose, for the sake of contradiction, that $W_i \cup W_j \subseteq V_\ell$ and a switch moves some vertex $v \in W_i \cup W_j$ to another district. Since W_i and W_j are leaves in T , only their cut vertices can be adjacent to vertices outside of $W_i \cup W_j$. Then, v is a cut vertex in $\Pi(v)$ and $\Pi(v) \setminus \{v\}$ does not induce a connected subgraph in G , a contradiction.

Since $W_i \neq W_j$, there are at least two vertices, one from each block, that remain in V_ℓ after any sequence of switch operation. Consequently, V_ℓ cannot become a singleton. \square

Lemma 3. *Let Π be a k -district map on G for some $k \geq 2$, and let $V_i \in \Pi$ such that V_i contains at most one leaf of the block tree T of G . Then V_i is shrinkable. Furthermore,*

- *if V_i does not contain any leaf of the block tree, then V_i can be shrunk to any of its vertices;*
- *if V_i contains a leaf $W_j \in B(G)$ of the block tree, then V_i can be shrunk to a vertex v if and only if $v \in W_j$ and v is not the parent cut vertex of W_j .*

In both cases, a sequence of $|V_i| - 1$ switches that shrink V_i can be computed in $O(|E(G)|)$ time.

Proof. We first prove a necessary condition for shrinking a district to a target vertex. Assume that V_i can be shrunk to a vertex $t \in V_i$, and V_i contains exactly one leaf $W_j \in B(G)$ of the block tree. Let c_j be the parent cut vertex of W_j . Since every path between $W_j \setminus \{c_j\}$ and $V_i \setminus W_j$ contains c_j , no vertex in $W_j \setminus \{c_j\}$ can change districts until c_j and all vertices of V_i outside of W_j have switched to some other districts. At this point, we have $V_i = W_j \setminus \{c_j\}$, consequently $t \in W_j \setminus \{c_j\}$, as required.

We next show that the above conditions are sufficient. Assume that V_i and a target vertex $t \in V_i$ satisfy the above restrictions. It is enough to show that if $V_i \neq \{t\}$, there exists a vertex $v \in V_i \setminus \{t\}$, such that v can be switched to another district; and t and $V_i \setminus \{v\}$ satisfy the conditions above. Then we can successively switch all vertices in $V_i \setminus \{t\}$ to other districts until $V_i = \{t\}$.

Let G' be the subgraph induced by V_i . Compute the block tree of G' , and denote it by T' . Root T' at the block vertex in the tree that contains t . We distinguish between cases.

- If G' is not biconnected, then G' contains two or more leaf blocks. Let $W' \in B(G')$ be a leaf block in T' other than the root, and let $c' \in C(G')$ be its parent cut vertex. Note that W' is not a leaf block in T , otherwise V_i would contain this leaf block, contradicting our assumptions. Then, it is either a subset of a nonleaf block of T or a proper subset of a leaf block of T . In either case, there exists a vertex $v \in W' \setminus \{c'\}$ adjacent to some vertex $u \notin V_i$. Since W' is biconnected, $W' \setminus \{v\}$ induces a connected subgraph in G ; consequently $V_i \setminus \{v\}$ induces a connected subgraph, as well. Therefore, v can be switched to the district of u .
- If G' is biconnected, then G' is a subgraph of some block $W \in B(G)$. We claim that there exists a vertex $v \in V_i \setminus \{t\}$ adjacent to some vertex $u \notin V_i$. To prove the claim, suppose the contrary. Then every path between $V_i \setminus \{t\}$ and $V(G) \setminus V_i$ goes through t . This implies that t is a cut vertex, and V_i is a leaf block in T , which contradicts our assumption. Now again, v can be switched to the district of u .

First, note that the switch operation maintain the property that V_i contains at most one leaf block of T . Indeed, since we shrink V_i , the number of leaf blocks contained in V_i monotonically decreases. Second, we show that t remains a valid choice for the target vertex. If $V_i \setminus \{v\}$ contains the same leaf blocks as V_i , then t remains a valid target. Otherwise V_i contains a leaf block, say W_j , and $V_i \setminus \{v\}$ does not, then v is the parent cut vertex of W_j . In this case, $t \in V_i \setminus \{v\}$, and any vertex in $V_i \setminus \{v\}$ is a valid choice for t . This proves that V_i is shrinkable, as required.

Our proof is constructive and leads to an efficient algorithm that successively switches every vertex in $V_i \setminus \{t\}$ to some other districts until $V_i = \{t\}$. The block trees T and T' can be computed in $O(|E(G)|)$ time [22]. While V_i is shrunk, we maintain the induced subgraph G' , and the set of edges between V_i and $V(G) \setminus V_i$ in $O(|E(G)|)$ total time. While T' contains two or more blocks, we can successively switch all vertices of a leaf block W' that does not contain t to other districts; eliminating the need for recomputing T' . Then, the total running time is $O(|E(G)|)$. \square

Lemma 4. *The shrinkability (resp., unshrinkability) of a k -district map on a graph G is invariant under switch operations.*

Proof. Every unshrinkable k -district map contains some unshrinkable district V_ℓ . Lemmas 2–3 show that a subset $V_\ell \in \Pi$ is unshrinkable if and only if V_ℓ contains at least two leaves of the block tree, say $W_i, W_j \subset V_\ell$. By Lemma 3, $W_i \cup W_j \subseteq V_\ell$ after any sequence of switches, so V_ℓ remains unshrinkable. The rest of the proof is implied by the reversibility of switches. \square

3 Connectedness

In this section we characterize graphs G for which the switch graph $\Gamma_k(G)$ is connected. We give two results depending on the connectivity of G .

3.1 Biconnected Graphs

Theorem 5. *For every biconnected graph G with n vertices, and for every integer $1 \leq k \leq n$, the switch graph $\Gamma_k(G)$ is connected and its diameter is bounded by $O(kn)$.*

Proof. We may assume that $1 < k < n$, otherwise $\Gamma_k(G)$ is trivially connected. We present an algorithm (Algorithm 1) that performs a sequence of switches that transform Π into a canonical k -district map of G , that we denote by Π_0 . We show that Π_0 depends only on G and k (but not on Π). Consequently, any two k -district maps can be transformed to Π_0 , and $\Gamma_k(G)$ is connected.

Algorithm 1 Canonical Algorithm for Biconnected Graphs

```

1: procedure CANONICAL( $G, k, \Pi$ )
2:   while  $k > 1$  do
3:     Compute the SPQR tree  $T_G$  of  $G$ ; order the leaves by DFS; let  $\mu$  be the first leaf.
4:     if  $\mu$  is an S node (and  $\text{skeleton}(\mu)$  is a cycle with one virtual edge) then
5:       Let  $\text{skeleton}(\mu) = (v_1, \dots, v_t)$ , where  $v_1v_t$  is the virtual edge; set  $i = 2$ .
6:       while  $i < t$  and  $k > 1$  do
7:         Shrink  $\Pi(v_i)$  to  $\{v_i\}$ .
8:         Delete vertex  $v_i$  from  $G$ , and put  $i := i + 1$  and  $k := k - 1$ .
9:       else  $\mu$  is an R node (and  $\text{skeleton}(\mu)$  is triconnected)
10:        Let  $v$  be an arbitrary vertex that is not incident to the (unique) virtual edge.
11:        Shrink  $\Pi(v)$  to  $\{v\}$ .
12:        Delete vertex  $v$  from  $G$ , and put  $k := k - 1$ .

```

Proof of Correctness. Algorithm 1 successively shrinks a district into a single vertex, and then deletes this vertex from the graph, and the corresponding district from Π , until the number of districts drops to 1. We need to show that each district that the algorithm shrinks into a singleton is shrinkable. We prove an invariant that imply this property:

Claim 6. *G remains connected and the district map Π remains shrinkable during Algorithm 1.*

Proof. In a biconnected graph, every district is shrinkable by Lemma 3. Let μ be the leaf node in line 3 of the algorithm. If μ is a R node, then the graph G remains biconnected after the deletion of a vertex, and so the $(k-1)$ -district map of the remaining graph is shrinkable. If μ is an S node, then G obtained by deleting vertex v_i is either biconnected or a biconnected graph with a “dangling” path (v_{i+1}, \dots, v_t) . In both cases, G has at most one leaf block (namely, a 1-edge block at the end of

the dangling path). By Lemma 3, every district that contains at most one leaf block is shrinkable, and so the district map remains shrinkable. \square

The following claim establishes that the switch graph $\Gamma_k(G)$ is connected since it contains a path from any district map to the district map produced by Algorithm 1.

Claim 7. *The district map Π_0 depends only on G and k .*

Proof. The map Π_0 contains the deleted singleton districts and one larger district. Since each vertex deleted from the graph G was selected based on the current graph G , its SPQR tree, and the DFS order of its leaves, the sequence of deleted vertices depends only on G and k . \square

Analysis. Algorithm 1 successively shrinks $k - 1$ districts into singletons. By Lemma 3, for each district this is done by a sequence of $O(n)$ switches that can be computed in $O(|E(G)|)$ time. Overall Algorithm 1 runs in $O(k|E(G)|)$ time and performs $O(kn)$ switch operations. For any two k -district maps, Π_1 and Π_2 , there exists a sequence of $O(kn)$ switches that takes Π_1 to Π_0 and then to Π_2 . Therefore, the diameter of $\Gamma_k(G)$ is $O(kn)$. \square

The following theorem shows that the upper bound in Theorem 5 is asymptotically tight.

Theorem 8. *For all integers $1 \leq k \leq n$, there exists a biconnected graph G with n vertices such that the diameter of $\Gamma_k(G)$ is $\Omega(k(n - k))$.*

Proof. Let $G = C_n$ be the cycle with n vertices (v_1, \dots, v_n) . We construct two k -district maps, Π_1 and Π_2 . Let Π_1 consist of $V_i = \{v_i\}$ for $i = 1, \dots, k - 1$, and $V_k = \{v_k, \dots, v_n\}$. The partition Π_2 is the copy of Π_1 rotated by $\lfloor n/2 \rfloor$, that is, $V'_i = \{v_{i+\lfloor n/2 \rfloor}\}$ for $i = 1, \dots, k - 1$, and $V'_k = \{v_{k+\lfloor n/2 \rfloor}, \dots, v_{n+\lfloor n/2 \rfloor}\}$, where we use arithmetic modulo n on the indices.

Assume that a sequence of switch operations takes Π_1 to Π_2 . Note that the cyclic order of the district cannot change, and so there is an integer $r \in \{0, \dots, k - 1\}$ such that V_i is transformed to $V'_{i+r \bmod k}$ for all $i \in \{1, \dots, k\}$. For any r , at least $k - 2$ districts are singletons in both Π_1 and Π_2 . The sum of the shortest distances along C_n between the initial and target positions is a lower bound for the number of switches.

If $r \leq \lfloor k/2 \rfloor$, then the shortest distance between the initial and target positions is at least $\lfloor n/2 \rfloor - r \in \Omega(n - k)$ for the districts V_i , $i = 1, \dots, k - 1 - r$; which sums to $\Omega(k(n - k))$. If $\lfloor k/2 \rfloor < r < k$, then shortest distance is at least $\lfloor n/2 \rfloor - (k - r) \in \Omega(n - k)$ for V_i , $i = r, \dots, k - 1$; which also sums to $\Omega(k(n - k))$. \square

3.2 Algorithm for General Connected Graphs

Recall that $\Gamma'_k(G)$ is the subgraph of $\Gamma_k(G)$ induced by shrinkable district maps. If G is a biconnected graph, then every district map is shrinkable by Lemma 3, and so $\Gamma_k(G) = \Gamma'_k(G)$. In this section, we extend this result to a larger family of graphs, showing that if G is connected, then $\Gamma'_k(G)$ is connected. That is, any shrinkable k -district map can be carried to any other shrinkable k -district by a sequence of switch operations.

Theorem 9. *For every connected graph G with n vertices, and for every integer $1 \leq k \leq n$, the switch graph $\Gamma'_k(G)$ over shrinkable k -district maps is connected and its diameter is $O(kn)$.*

A crucial technical step is to move a district from one block to another, through a cut vertex. This is accomplished in the following technical lemma.

Lemma 10. *Let G be a connected graph whose block tree contains at least two blocks, $W_1, W_2 \in B(G)$, and let P be a shortest path from a vertex in W_1 to a vertex in W_2 (possibly, P has a single vertex). Let Π be a district map of G in which each vertex of P is a singleton district, but W_1 contains a district of size more than one. Then there is a sequence of $O(|W_1| + |P|)$ switches that increases the number of districts in W_1 by one, and decreases the number districts in W_2 by one.*

Proof. Let $c_1 \in W_1$ and $c_2 \in W_2$ be the two endpoints of P ; possibly $c_1 = c_2$. Note that $c_1, c_2 \in C(G)$ since P is a shortest path between W_1 and W_2 . We claim that after $O(|W_1|)$ switch operations in W_1 , we can find a path $P^* = (p_0, p_1, \dots, p_m)$ such that $\{p_0, p_1\}$ is a 2-vertex district in W_1 , all other vertices in P^* are singleton districts, and $P = (p_\ell, \dots, p_m)$ for $1 \leq \ell \leq m$ (with $p_\ell = c_1$ and $p_m = c_2$). Assuming that this is possible, we can then successively perform $\text{switch}(p_{i-1}, p_i, p_{i+1})$ for $i = 1, \dots, m-1$, which replaces $\{p_0, p_1\}$ by two singleton districts, and produces a 2-vertex district $\{p_{m-1}, p_m\}$. Finally, we shrink this district to $\{p_{m-1}\}$ by Lemma 3, thereby decreasing the number of districts in W_2 by one. Overall, we have used $O(|W_1| + |P^*|) = O(|W_1| + |P|)$ switches.

To prove the claim, let G_1 be the biconnected subgraph of G induced by W_1 . Let $Q = (q_1, \dots, q_s)$ be a shortest path between $q_s = c_1$ and a vertex in a district $V_0 \subseteq W_1$ of size $|V_0| > 1$. Since Q is a shortest path, the vertices q_2, \dots, q_s are singleton districts. If $|V_0| = 2$, say $V_0 = \{q_0, q_1\}$, then we can take $P^* = (q_0, q_1, \dots, q_s) \oplus P$, where \oplus is the concatenation operation.

Assume that $|V_0| > 2$. Since G_1 is biconnected, V_0 can be shrunk to $\{q_1\}$ by a sequence of $|V_0| - 1 = O(|W_1|)$ switches by Lemma 3. Each switch in the sequence shrinks V_0 and expands some adjacent district. Perform the switches in this sequence until either (a) $|V_0| = 2$, or (b) some singleton district $\{q_i\}$, $i = 2, \dots, s$, expands. In both cases, we find a path $Q' = (q_i, \dots, q_s)$, $i \in \{1, \dots, s\}$, such that q_i is in some 2-vertex district $\{q_0, q_i\}$, all other vertices in Q' are singletons, and $q_s = c_1$. Consequently, we can take $P^* = (q_0, q_i, \dots, q_s) \oplus P$, as claimed. \square

We can now consider the general case. Let G be a connected graph with n vertices and let $1 \leq k \leq n$. We present an algorithm (Algorithm 2) that transforms a given shrinkable k -district map Π into one in pseudo-canonical form (defined below), and then show that any two k -district maps in pseudo-canonical form can be transformed to each other. Consequently, any two shrinkable k -district maps can be transformed into each other, and $\Gamma'_k(G)$ is connected.

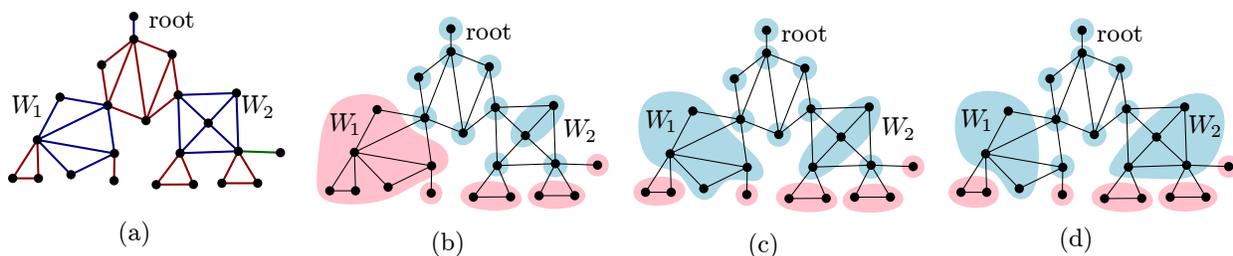


Figure 2: (a) A connected graph with nine blocks. (b) A pseudo-canonical 15-district map. Five leaf districts are red and ten nonleaf districts are blue. (c) and (d): Pseudo-canonical district maps obtained from (b) by moving districts from W_2 to W_1 by successive applications of Lemma 10.

We introduce some additional terminology; see Figure 2 (a). Let T be a block tree of G . Fix an arbitrary leaf block $R \in B(G)$. We consider T as an ordered tree, rooted at R , where the children of each node are ordered arbitrarily. For a district map Π , we define a **leaf district** to be a district that contains every vertex of some nonroot leaf block $W \in B(G)$, with the possible exception of its parent cut vertex $c \in C(G)$. Note that a leaf district could have vertices outside the

leaf block. Moreover, every leaf district V_i corresponds to a unique leaf block (otherwise Π would be unshrinkable by Lemma 2), and we denote this block by $\text{leaf}(V_i)$. A leaf district is shrinkable into any vertex in $\text{leaf}(V_i)$, except for c (cf. Lemma 3). Further note that a district may become a leaf district over the course of the algorithm, while leaf districts remain leaf districts.

For every block $W \in B(G)$, except for the root, we define a set $\text{down}(W)$ as follows. Let $c \in C(G)$ be the parent of W in T , let V_i be the district that contains c , and let $\text{down}(W)$ be the set of vertices in V_i that lie in W or its descendants. The set $\text{down}(W)$ is an **elbow** if $\text{down}(W) \neq \{c\}$, V_i is a leaf district, and $\text{down}(W)$ does not contain the block $\text{leaf}(V_i)$; see Figure 3. An elbow is **maximal** if it is not contained in another elbow. A leaf district is **elbow-free** if it does not contain any elbows.

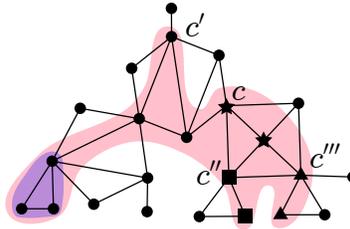


Figure 3: Example for the definitions of $\text{down}(\cdot)$ and elbow. District V_i (pink) contains a leaf block (purple). Let W , W' , W'' , and W''' be the blocks whose parent cut vertices are c , c' , c'' , and c''' , resp. The set $\text{down}(W') = V_i$ is not an elbow since it contains $\text{leaf}(V_i)$. Vertices shown as squares (triangles) are in $\text{down}(W'')$ ($\text{down}(W''')$). The vertices in $\text{down}(W)$ are shown as stars, squares and triangles. Sets $\text{down}(W)$, $\text{down}(W'')$ and $\text{down}(W''')$ are elbows while only the first is maximal.

A district map of G is in **pseudo-canonical** form if every block $W \in B(G)$ satisfies one of the following three mutually exclusive conditions (see Figure 2 for examples):

- (i) all vertices in W are in singleton nonleaf districts;
- (ii) all vertices of W , with the possible exception of the parent cut-vertex of W , are in the same leaf district. Moreover, if W is not a leaf block, then this district contains the leftmost grandchild block of W .
- (iii) all vertices of W are in nonleaf districts, whose vertices are all contained in W , but not all are singletons, and all ancestor (resp., descendant) blocks of W are of type (i) (resp., type (ii));

We refer to the condition that a block satisfies as its **type**. Notice that (iii) implies that blocks of type (i) (or blocks of types (i) and (iii)) induce a connected subtree of T containing the root. The proof of Theorem 9 is the combination of Lemmas 11 and 12.

Lemma 11. *Let G be a connected graph with n vertices and let $1 \leq k \leq n$. Every shrinkable k -district map can be taken into pseudo-canonical form by a sequence of $O(kn)$ switches.*

Proof. Let Π be a shrinkable k -district map. Algorithm 2 (below) transforms Π into pseudo-canonical form in three phases; refer to Figure 4. Each phase processes all blocks in $B(G)$ in DFS order of the block tree T . Phase 1 eliminates elbows. Phase 2 shrinks leaf districts such that they are each confined to their leaf blocks. Phase 3 shrinks all nonleaf districts to singletons (or possibly turns some nonleaf districts into leaf districts). We continue with the details.

In lines 3 and 7, Algorithm 2 shrinks $\text{down}(W)$ and $W \cap V_i$, resp., into a singleton $\{c\}$. We describe these subroutines here. In both cases, we invoke Lemma 3 for a district map Π' on a subgraph G' of G ; where Π' is the restriction of Π to G' . In the first case, G' is induced by c and

its descendants. Note that $\text{down}(W)$ is a district in Π' , and it is shrinkable by Lemma 3 since $\text{down}(W)$ is an elbow in G and contains no leaf districts. In the second case, G' is obtained from G by deleting all descendants of c . Now $W \cap V_i$ is a district in Π' since V_i is a leaf district in Π , W is the highest nonleaf block (in DFS order) that intersects V_i , and V_i does not contain elbows by invariant (II) below. By Lemma 3, $W \cap V_i$ is shrinkable as it lies in a single block W . In both cases, Lemma 3 yields a sequence of switches that shrink $\text{down}(W)$ and $W \cap V_i$, resp., to $\{c\}$.

In lines 13 and 16, Algorithm 2 shrinks a district V_i with $c' \in V_i$ to $\{c'\}$. In both cases, V_i is shrinkable to c' by Lemma 3, and the proof of Lemma 3 provides an algorithm that successively switches vertices in $V_i \setminus \{c'\}$ to other districts arbitrarily. However, this process might introduce a new elbow. Here, we specify a particular a sequence of switches to ensure that no new elbows are created. While V_i is not a singleton, identify a noncut-vertex v of V_i adjacent to a vertex w in a nonleaf district V_j . (For example, see Figure 5(c)-(d) where W_1 has the role of W' .) If no such vertex exists, choose v adjacent to a vertex w in a leaf district V_j that intersects the leftmost grandchild block of W' . (For example, see Figure 5(e)-(f) where W_1 has the role of W' .) Let u be a neighbor of v in V_i , and apply $\text{switch}(u, v, w)$.

Algorithm 2 Pseudo-Canonical Algorithm for Connected Graphs

```

1: procedure PSEUDO-CANONICAL( $G, k, \Pi$ )
2:   for every nonroot block  $W \in B(G)$  in DFS order of  $T$  do
3:     if  $\text{down}(W)$  is an elbow then let  $c \in C(G)$  be  $W$ 's parent, shrink  $\text{down}(W)$  to  $\{c\}$ .
4:   for every nonleaf block  $W \in B(G)$  in DFS order of  $T$  do
5:     if  $W$  intersects a leaf district, then
6:       for each leaf district  $V_i$  that intersects  $W$  do
7:         shrink  $W \cap V_i$  onto the cut-vertex  $c$  of  $W$  in the descending path of  $T$  to  $\text{leaf}(V_i)$ ;
8:         apply an additional switch to contract  $V_i$  out of the block  $W$ .
9:   for every block  $W \in B(G)$  in DFS order of  $T$  do
10:    while  $W$  satisfies neither (i) nor (ii), and a grandchild  $W'$  of  $W$  is not of type (ii) do
11:      if  $W$  is the root of  $T$  and  $W$  is contained in a single district, then
12:        Let  $c'$  be a noncut-vertex of  $W$ , and let  $V_i$  be the district containing  $c'$ ;
13:        Shrink  $V_i$  to  $\{c'\}$ .
14:      else
15:        Let  $c'$  be the parent cut-vertex of  $W'$ , and let  $V_i$  be the district containing  $c'$ ;
16:        Shrink  $V_i$  to  $\{c'\}$ .
17:        if  $W'$  is still not of type (ii) and  $W$  is not of type (i), then
18:          Use Lemma 10 with  $P = (c')$  to move a district from  $W'$  to  $W$ .
19:      if  $W$  still satisfies neither (i) nor (ii), then
20:        Shrink the district containing the parent cut-vertex  $c$  of  $W$  to  $\{c\}$ .

```

Analysis of Algorithm 2. Note that maximal elbows are pairwise disjoint, and every block intersects at most one maximal elbow (by the definition of $\text{down}(W)$).

Phase 1 (lines 2-3) iterates over all nonroot blocks. In the course of Phase 1, we maintain the invariant that if W has been processed, then $\text{down}(W)$ is not an elbow. When the for-loop reaches a block W where $\text{down}(W)$ is an elbow, then it is a maximal elbow due to the DFS traversal of T , and $\text{down}(W)$ is shrunk to a cut vertex c , and produces $\text{down}(W) = \{c\}$, which is not an elbow. We also show that this does not create any new elbows. Indeed, if a switch shrinks $\text{down}(W)$ out of a cut vertex c' , then c' is a descendant of c , and some district V_j that intersects a child block W' of c' expands into c' . At this time, c' becomes the highest vertex of V_j , and so $\text{down}(W')$ contains

leaf(V_j) if V_j is a leaf district (hence $\text{down}(W')$ cannot be an elbow). Thus, we conclude that Phase 1 successively eliminates all elbows and does not create any new elbow. Since the maximal elbows are pairwise disjoint, the sum of their cardinalities is at most n , and they can be shrunk with $O(n)$ switches. In Phases 2-3, we maintain invariant (I1): There are no elbows in the district map.

Phase 2 (lines 4-8) is a for-loop over all nonleaf blocks. In the course of Phase 2, we maintain the invariant that if W has been processed, then W is disjoint from leaf districts. When the for-loop reaches a block W that intersects a leaf district V_i , then V_i has no elbows by invariant (I1), and the ancestors of W are disjoint from V_i (because we visit blocks in DFS order). Consequently $V_i \cap W$ is shrinkable to the child of W that leads to the leaf block $\text{leaf}(V_i)$. For each leaf district V_i , Phase 2 uses $O(n)$ switches to shrink V_i , and $O(kn)$ switches overall. No elbows are created since leaf districts are never expanded to a block they do not already intersect (with the possible exception of the parent cut vertex of a block). In Phase 3, we maintain invariant (I2): If a leaf district intersects a block, then such block is of type (ii).

Phase 3 (lines 9-20) is a for-loop over all blocks $W \in B(G)$; see Figure 5 for an example of the execution of this phase. In the course of Phase 3, we maintain the invariant that if W has been processed, it satisfies condition (i), (ii), or (iii) in the definition of pseudo-canonical forms. Indeed, for every block W , the switch operations modify only W or its descendants. The fact that we are processing W means that its grandparent is of type (i) when we begin processing W . Then, W intersects more than one district and we can shrink V_i to $\{c'\}$ in line 16 without expanding any districts not contained in W and in ancestors of W . This already implies that (I1) is maintained. Furthermore, if W satisfies conditions (i) or (ii), then the districts in W remain unchanged. Otherwise, the while-loop (lines 10-18) ensures that every district that intersects W is contained in W . In each iteration of the while-loop, V_i is a nonleaf district by (I2), and V_i is contained in the union of W and its descendants. The switches in lines 13 and 16 do not decrease the number of districts in W . The preference of expansions to shrink V_i to $\{c'\}$ ensures that (I2) is maintained for leaf districts. Indeed, such a switch may expand a leaf district if there is no other option: in this case V_i contains an entire block W^* , which is a descendant of W and whose grandchildren are of type (ii); after shrinking V_i to the parent cut-vertex of W^* expanding a leaf district, W^* becomes of type (ii). Using Lemma 10 in line 18 ensures that, eventually, W is of type (i) or (ii), or all its grandchildren are of type (ii). Finally, when the while loop terminates, lines 19-20 ensure that the parent cut vertex of W is a singleton, and so all ancestors of W comprise singletons. In Phase 3, $O(n)$ switches shrink each district, amounting to $O(kn)$ switches overall.

We have shown that Algorithm 2 takes any input district map Π into pseudo-canonical form. The three phases jointly use $O(kn)$ switches, as claimed. \square

We now introduce a method to transform a pseudo-canonical k -district map into another.

Lemma 12. *Let G be a connected graph with n vertices and let $k \leq n$ be a positive integer. For any two pseudo-canonical k -district maps, Π_1 and Π_2 , there is a sequence of $O(kn)$ switches that take Π_1 to Π_2 .*

Proof. Our proof is constructive: for a given district map Π in pseudo-canonical form, we assign every leaf district to the unique leaf block it intersects, and assign every nonleaf district to the highest (closest to the root) block in T it is contained in. For every block $W \in B(G)$, let $d_\Pi(W)$ be the number of districts assigned to W in Π . Notice that $\sum_{W \in B(G)} d_\Pi(W) = k$.

First, we explain how to transform Π_1 into an intermediate pseudo-canonical district map Π_m so that $d_{\Pi_m}(W) = d_{\Pi_2}(W)$ for every block $W \in B(G)$.

Suppose that $d_{\Pi_1}(W) \neq d_{\Pi_2}(W)$ for some block $W \in B(G)$ (otherwise, we can trivially set $\Pi_m = \Pi_1$). Since $\sum_{W \in B(G)} d_{\Pi_1}(W) = \sum_{W \in B(G)} d_{\Pi_2}(W) = k$, there exist blocks $W_1, W_2 \in B(G)$ such that $d_{\Pi_1}(W_1) < d_{\Pi_2}(W_1)$ and $d_{\Pi_1}(W_2) > d_{\Pi_2}(W_2)$.

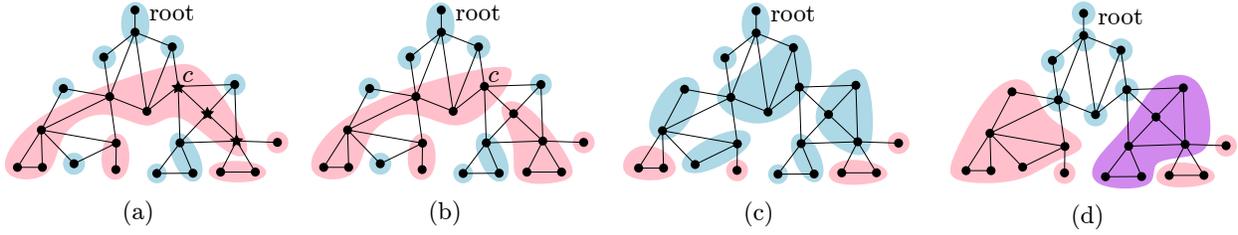


Figure 4: (a) A 12-district map of a graph. The four leaf districts are red, eight nonleaf districts are blue; c is the highest cut vertex in an elbow whose vertices are shown as stars. (b), (c), and (d) show the result of Phases 1, 2, and 3 of Algorithm 2, respectively. In Phase 3, a nonleaf district becomes a leaf district (shaded purple).

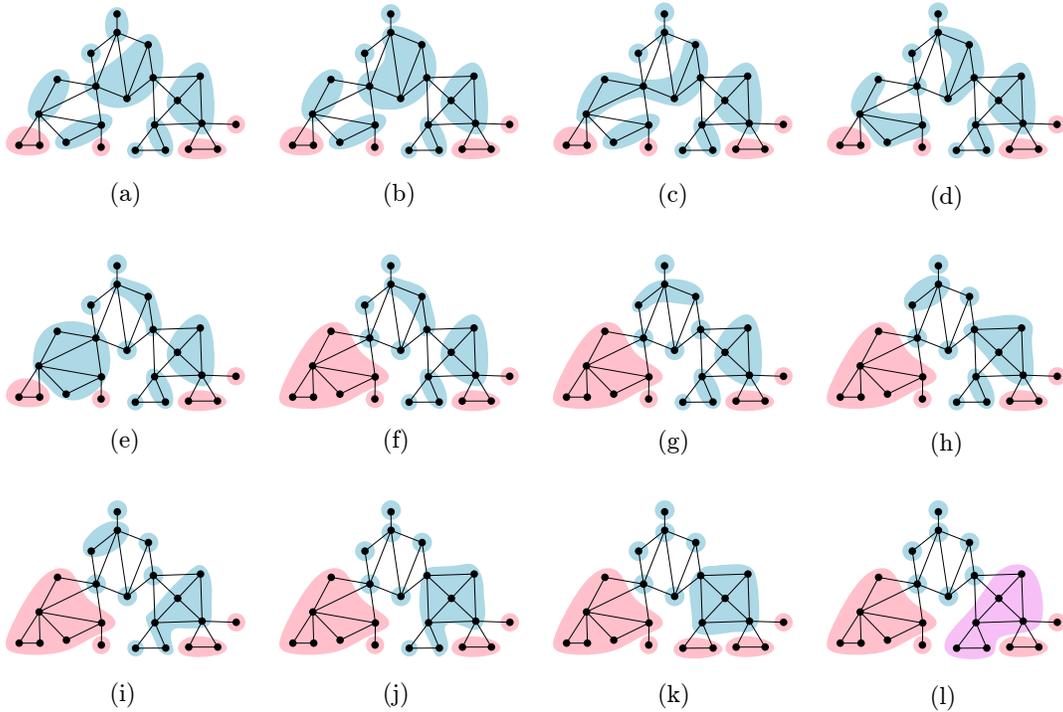


Figure 5: Breakdown of the example of Phase 3 from Figure 4(c) to Figure 4(d). The condition in line 10 is satisfied in (a) and (b) where W is the root block, (c), (e), (f), (h) and (j) where W is the grandchild of the root block. Only in (a) the conditions in line 11 are satisfied. Lemma 10 (line 18) is applied in (d), (g), and (i). While shrinking a district V_i in Figure 5(j), a nonleaf district becomes a leaf district in Figure 5(k). Continuing the shrinking, causes W_2 to become type (ii) in Figure 5(l).

Claim 13. *Let W_1 (resp., W_2) be a highest (resp., lowest) block such that $d_{\Pi_1}(W_1) < d_{\Pi_2}(W_1)$ (resp., $d_{\Pi_1}(W_2) > d_{\Pi_2}(W_2)$), then all ancestor blocks of W_1 and W_2 are of type (i) in Π_1 , and all descendant blocks of W_1 and W_2 are of type (ii) in Π_1 .*

Proof. Notice that if a block is of type (i) (resp., type (ii)) then it has been assigned with the maximum (resp., minimum) number of districts that it can possibly be assigned to. Then, W_1 cannot be of type (i) in Π_1 and it cannot be of type (ii) in Π_2 . By the definition of pseudo-canonical forms, all descendant (resp., ancestor) blocks of W_1 are of type (ii) (resp., type (i)) in Π_1 (resp., Π_2). By the choice of W_1 , all ancestor blocks of W_1 are of type (i) in Π_1 . An analogous argument proves the claim for W_2 . \square

Next, we construct an intermediate district map Π_m by successively reducing the difference in the d functions. While $d_{\Pi_1} \neq d_{\Pi_2}$, we transform Π_1 into another district map Π'_1 in pseudo-canonical form such that

$$\sum_{W \in B(G)} |d_{\Pi'_1}(W) - d_{\Pi_2}(W)| < \sum_{W \in B(G)} |d_{\Pi_1}(W) - d_{\Pi_2}(W)|. \quad (1)$$

Let W_1 and W_2 be blocks chosen as in Claim 13, let c_1 and c_2 be their respective parent cut-vertices, and let P be a shortest path between c_1 and c_2 . By Claim 13, all blocks along P are of type (i) in Π_1 , and so every vertex in P is in a singleton district. Applying Lemma 10 to Π_1 , we can move a district from W_2 to W_1 using $O(|W_1| + |P|) \leq O(n)$ switches. We need to make sure that the new map is also in pseudo-canonical form. If $d_{\Pi_1}(W_1) = 0$ (i.e., W_1 is of type (ii), but not a leaf block) shrink the leaf district out of W_1 by expanding the new nonleaf district that has moved into W_1 . If W_2 consists of a single (nonleaf) district, shrink it onto $\{c_2\}$ while expanding the leaf district of its leftmost grandchild W'_2 . The number of districts assigned to a block changes only in W_1 , W_2 , and (possibly) W'_2 . The procedure described above increases $d(W_1)$ by one, and decreases $d(W_2)$ (and possibly $d(W'_2)$) by one, making the difference smaller as claimed. The type of W_1 (resp., W_2) becomes (i) or (iii) (resp., (iii) or (ii)) and, by Claim 13, Π'_1 is in pseudo-canonical form.

In summary: while $d_{\Pi_1} \neq d_{\Pi_2}$, we repeat the above procedure. When the while loop ends, we find a pseudo-canonical district map Π_m such that $\sum_{W \in B(G)} |d_{\Pi_m}(W) - d_{\Pi_2}(W)| = 0$ (and thus $d_{\Pi_m} = d_{\Pi_2}$). Initially $\sum_{W \in B(G)} |d_{\Pi_1}(W) - d_{\Pi_2}(W)| \leq 2k$ and each step decreases the difference by at least one, and so at most $2k$ iterations will be needed. Since each iteration takes $O(n)$ switches, this process uses $O(nk)$ switches overall.

In order to complete the proof of Lemma 12, we need to show how to reconfigure Π_m to Π_2 . Recall that both district maps are in pseudo-canonical form and they satisfy $d_{\Pi_m} = d_{\Pi_2}$. Further, if a district map is in pseudo-canonical form, each block is of one of three possible types. We claim that every block of G is of the same type in both Π_m and Π_2 . For ease of notation, we assume $\Pi_1 = \Pi_m$. If W is the root and $d_{\Pi_i}(W) = |W|$, $i \in \{1, 2\}$, or W is a nonroot block and $d_{\Pi_i}(W) = |W| - 1$, then W is of type (i). If W is a leaf block and $d_{\Pi_i}(W) = 1$, or W is a nonleaf block and $d_{\Pi_i}(W) = 0$ then W is of type (ii). Else, W is of type (iii). This implies that every block of type (i) consists of singletons; and the union of blocks of type (ii) are partitioned identically into leaf districts in both Π_1 and Π_2 since, by definition of type (ii), the leaf district that intersects the block must contain the leftmost grandchild block, and by the fact that there are no elbows. Thus, no switches are required in these blocks. Blocks of type (iii) each contain the same number of districts in both Π_1 and Π_2 . These blocks are pairwise disjoint by definition and all districts that intersect such a block is entirely contained in that block. Applying Algorithm 1 to each block W of type (iii), both Π_1 and Π_2 transform to the same district map in $O(k|W|)$ switches by Theorem 5. Overall, this takes $O(kn)$ switches, completing the proof of Lemma 12. \square

3.3 Characterization of Connected Switch Graphs

Using Lemmas 2–4 and Theorem 9, we can characterize the pairs (G, k) , of a connected graph G and a positive integer k , for which the switch graph $\Gamma_k(G)$ is connected (cf. Theorem 17 below).

Lemma 14. *For a connected graph G with n vertices and an integer $1 \leq k \leq n$, the switch graph $\Gamma_k(G)$ is connected if and only if $k = 1$ or every k -district map is shrinkable (i.e., $\Gamma_k(G) = \Gamma'_k(G)$).*

Proof. The case that $k = 1$ is trivial, as $\Gamma_k(G)$ is a singleton. Assume $k \geq 2$ for the remainder of the proof. If every k -district map is shrinkable (i.e., $\Gamma_k(G) = \Gamma'_k(G)$), then $\Gamma'_k(G)$ is connected by Theorem 9, and so $\Gamma_k(G)$ is connected. If some k -district maps are shrinkable and some are unshrinkable, then $\Gamma_k(G)$ is disconnected, since there is no edge between the set of shrinkable and unshrinkable district maps by Lemma 4.

Finally, assume that every k -district map is unshrinkable in G . We show that $\Gamma_k(G)$ is disconnected. Let Π_1 be an arbitrary k -district map. By Lemmas 2–3, some district $V_i \in \Pi_1$ contains two leaf blocks of the block graph, say $W_a, W_b \in B(G)$, with parent cut vertices $c_a, c_b \in C(G)$ (possibly $c_a = c_b$). Since G is connected and $k \geq 2$, there exists a district V_j adjacent to V_i . We construct a k -district map Π_2 from Π_1 by replacing V_i and V_j with $V'_i := W_a \setminus \{c_a\}$ and $V'_j := (V_i \cup V_j) \setminus V'_i$. Importantly, none of the districts in Π_2 contain both W_a and W_b ; and by Lemma 2, every sequence of switch operations transforms V_i to a district that contains both W_a and W_b . Thus $\Gamma_k(G)$ does not contain any path between Π_1 and Π_2 , as required. \square

Lemma 2 allows us to efficiently check whether a connected graph G admits an unshrinkable k -district map. Let G be connected but not biconnected. For two leaf blocks $W_1, W_2 \in B(G)$, let $P(W_1, W_2)$ denote the **union** of W_1, W_2 , and the set of vertices along a shortest path in G between W_1 and W_2 . Let $M = \min\{|P(W_1, W_2)| : W_1, W_2 \in B(G) \text{ leaf blocks}\}$.

Lemma 15. *Let G be a connected graph with n vertices that is connected but not biconnected, and let $k \leq n$ be a positive integer. Every k -district map in G is shrinkable if and only if $n - k \leq M$.*

Proof. If $n - k \leq M$, then every district in a k -district map contains fewer than M vertices. By the definition of M , none of these districts can contain two leaf blocks, and, therefore, are shrinkable.

If $M > n - k$, then we construct a k -district map for G in which one of the districts is unshrinkable. Let $\widehat{V} \subset V(G)$ be a vertex set of minimum cardinality that contains two leaf blocks in $B(G)$ and a shortest path between them. By definition, we have $|\widehat{V}| = M$. By partitioning $V(G) \setminus \widehat{V}$ into singletons, we obtain a \widehat{k} -district map $\widehat{\Pi}$, where $\widehat{k} = n - M + 1$, and $\widehat{V} \in \widehat{\Pi}$. Successively merge pairs of adjacent districts until the number of districts drops to k (recall that G is connected, so some pair of districts are always adjacent). We obtain a k -district map Π , where one of the districts contains \widehat{V} , and is unshrinkable by Lemma 2, as required. \square

Lemma 16. *We can compute the value M in $O(n + m)$ time, where $n = |V(G)|$ and $m = |E(G)|$.*

Proof. Given a connected graph $G = (V, E)$, first compute the block tree, and modify G as follows: replace each leaf block by a path with the same number of vertices, such that one endpoint is the original cut vertex (and hence the other endpoint is a leaf), and denote by G' the resulting graph. Then we run a modified multi-source BFS on G' , starting from the leaves. The algorithm assigns two labels to every vertex $v \in V(G')$, the **level** $\ell(v)$ and a **cluster** $c(v)$. Initially, each leaf $v \in V(G')$ is assigned level $\ell(v) = 0$ and clusters $c(v) = v$. When the BFS visits a new vertex v along an edge uv , it sets $\ell(v) := \ell(u) + 1$ and $c(v) := c(u)$. Clearly, $\ell(v)$ is the distance from v to the closest leaf in G' , and $c(v)$ is one such leaf. After the BFS termination, our algorithm finds an edge $uv \in E(G')$ such that $c(u) \neq c(v)$ and $\ell(u) + \ell(v)$ is minimal, and returns $\ell(u) + \ell(v) + 2$.

The modified BFS runs in $O(n+m)$ time, and a desired edge uv can be found in $O(m)$ additional time, so the overall running time is $O(n+m)$. It remains to prove that $M = \ell(u) + \ell(v) + 2$. Note that u and v are at distance $\ell(u)$ and $\ell(v)$, resp., from the leaves $c(u)$ and $c(v)$. The cluster of $c(u)$ (resp., $c(v)$) contains a shortest path from u to $c(u)$ (resp., from v to $c(v)$), and so these shortest paths are disjoint. The concatenation of the two shortest paths is a shortest path P' between the leaves $c(u)$ and $c(v)$, and it has $\ell(u) + \ell(v) + 2$ vertices. The path P' contains the chains incident to u and v in G' . By the definition of G' , these chains correspond to leaf blocks W_1 and W_2 of the same size in G . Consequently, $\ell(u) + \ell(v) + 2 = P(W_1, W_2)$. Therefore, $M \leq \ell(u) + \ell(v) + 2$.

Conversely, assume that $M = P(W_1, W_2)$ for some leaf blocks $W_1, W_2 \in B(G)$. These leaf blocks correspond to chains ending in two leaves, say W'_1 and W'_2 , in G' . By construction, the distance between W'_1 and W'_2 is $d_{G'}(W'_1, W'_2) = M - 1$. Let P' be a shortest path between W'_1 and W'_2 . We claim that for every vertex v' in P' , $\ell(v')$ is the minimum distance to $\{W'_1, W'_2\}$, i.e., $\ell(v') = \min\{d_{G'}(v', W_1), d_{G'}(v', W_2)\}$. Suppose, to the contrary, that there is a vertex v' in P' for which $\ell(v') \neq \min\{d_{G'}(v', W'_1), d_{G'}(v', W'_2)\}$. Since $\ell(v')$ is the minimum distance to some leaf in G' , we have $\ell(v') = d_{G'}(v', W'_3)$ for a leaf W'_3 , and $\ell(v') < \min\{d_{G'}(v', W'_1), d_{G'}(v', W'_2)\}$. As v' is in the path P' , $d_{G'}(v', W'_1) + d_{G'}(v', W'_2) = d_{G'}(W'_1, W'_2) = M - 1$. By the triangle inequality, $d_{G'}(W'_1, W'_3)$ or $d_{G'}(W'_2, W'_3)$ is less than $d_{G'}(W'_1, W'_2)$, contradicting the minimality of $P(W_1, W_2)$. Now P' contains two consecutive vertices, say u^* and v^* , such that $\ell(u^*) = d_{G'}(u^*, W'_1)$, $\ell(v^*) = d_{G'}(v^*, W'_2)$, and $c(u^*) \neq c(v^*)$. The sum of their distances to the two endpoints of P' is $\ell(u^*) + \ell(v^*) = (M - 1) - 1 = M - 2$, hence $M = \ell(u^*) + \ell(v^*) + 2$. Then, $\ell(u) + \ell(v) + 2 \leq M$, as required. \square

The combination of Theorem 5 and Lemmas 14–16 yields the following result.

Theorem 17. *For a connected graph G with n vertices and a positive integer $k \leq n$, the switch graph $\Gamma_k(G)$ is connected if and only if G is biconnected or $k + M \geq n$, which can be tested in $O(n+m)$ time, where $m = |E(G)|$.*

4 PSPACE-Completeness for Connectedness

In the **connectedness problem**, we are given a graph G , and two k -district maps, Π_A and Π_B , for some integer $1 \leq k \leq n$, and ask whether Π_A and Π_B are in the same component of the switch graph $\Gamma_k(G)$. In this section, we show that this problem is PSPACE-complete. Further, we show that the problem remains PSPACE-complete even if (i) we restrict G to be a planar graph of maximum degree 6, or (ii) we restrict the number of districts to $k = 2$. As an immediate consequence, we show that the diameter of a connected component of $\Gamma_k(G)$ may be as large as $2^{\Omega(n)}$ where $n = |V(G)|$. Membership in PSPACE is justified by the fact that a nondeterministic machine can explore $\Gamma_k(G)$ storing one district map at a time, so we focus now on proving hardness.

4.1 PSPACE-Hardness for General Graphs with Many Districts

We prove PSPACE-hardness by a reduction from the reconfiguration problem for Nondeterministic Constraint Logic (abbreviated NCL), which is known to be PSPACE-complete [20]. In this problem, we are given an **NCL graph**, which is a planar cubic graph where each edge is colored either blue or red, and each vertex is either an **OR** vertex incident on 3 blue edges, or an **AND** vertex incident on 2 red edges and 1 blue edge. An NCL graph with an orientation assigned to its edges is considered **satisfied** if all of its vertices are satisfied; an OR vertex is **satisfied** when at least one edge is oriented towards it, and an AND vertex is **satisfied** when both of its red edges are oriented towards it or its blue edge is oriented towards it. In the **NCL reconfiguration problem**, we are given an initial and a final orientation that both satisfy an NCL graph, and must decide whether one can

be reconfigured into the other by flipping the orientation of one edge at a time in such a way that after each flip the NCL graph is satisfied.

Given an NCL graph G_{NCL} , we create a graph G as follows. First create OR and AND gadgets of 7 and 9 vertices, respectively. The adjacencies between the vertices are shown in Figure 6. Given a vertex $v \in G_{NCL}$, we denote the corresponding gadget $F(v)$. For each gadget, the labelled vertices in Figure 6 are **terminals**, and the unlabelled vertices that are adjacent to the leaves are called **anchors**.

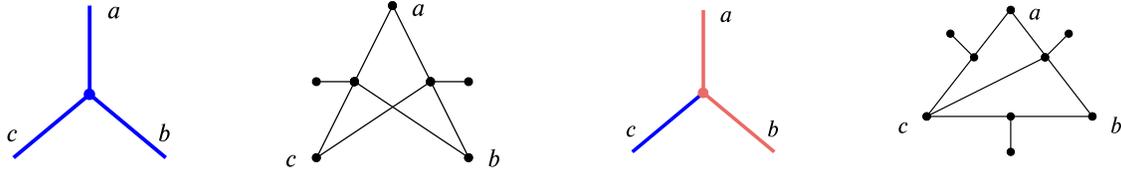


Figure 6: Gadgets for OR and AND vertices (left and right, respectively)

For the AND gadget, we call the two degree-two terminals (a and b) **red terminals**, and the degree-three terminal (c) a **blue terminal**. Each edge of G_{NCL} corresponds to a terminal in two gadgets, one for each vertex that edge is incident to (as shown by the labels in Figure 6), and thus we identify terminals of different gadgets that correspond to the same edge (Figure 7). This concludes the construction of G .

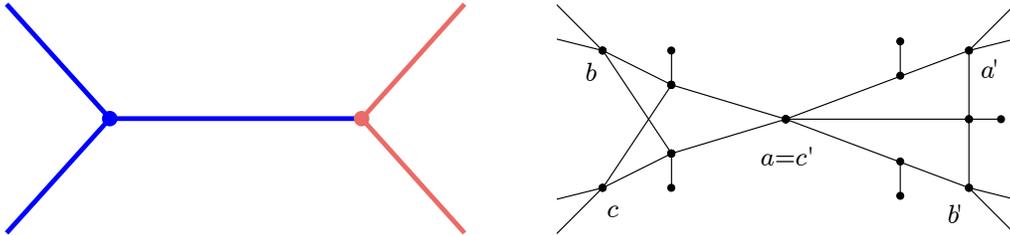


Figure 7: Two gadgets glued together along a shared terminal.

It remains to construct an initial and a final district map for G to simulate the initial and final orientations in G_{NCL} . Given an orientation O on G_{NCL} , construct a district map Π_O for G as follows: for every vertex $v \in G_{NCL}$, we construct a district that will contain most vertices of the gadget $F(v)$. This district always contains all anchors and leaves of that gadget. In addition, it contains the terminal corresponding to every edge e orientated towards v in O (see Figure 8). Note that every district in the initial or the final configuration contains at least two leaves. By Lemma 2, these leaves and their anchors cannot be moved to another district by any sequence of switch operations. Thus each district is tied to its respective gadget, that is,

- (\star) there is a one-to-one correspondence between the districts and the gadgets that remains invariant under 1-switch operations.

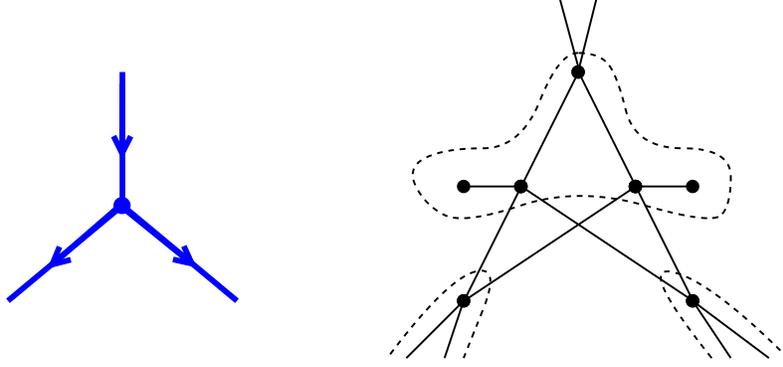


Figure 8: An orientation of an OR vertex and its associated district map

Lemma 18. *The district map Π_O over G is well defined if and only if the orientation O on G_{NCL} is satisfying.*

Proof. Consider an AND vertex and its associated gadget. We know by property (\star) , that the district corresponding to this gadget must contain three anchor vertices and three leaves. In order for the district to be connected it must contain (i) either the terminal associated with the blue edge (c in Figure 6), or (ii) both terminals associated with red edges (a and b in Figure 6). This is equivalent to saying that the AND vertex is satisfied by O .

Similarly, for the gadget associated to an OR vertex, its corresponding district has two anchors and two unlabelled leaves. In order for these vertices to remain connected within the district, the district must contain any of the three terminals of the gadget. This is equivalent to saying that O satisfies the OR vertex. \square

Lemma 19 (Flip-Switch Equivalence). *For every district map Π_{O_1} on G obtained from an orientation O_1 on G_{NCL} , every 1-switch operation on Π_{O_1} yields a map Π such that $\Pi = \Pi_{O_2}$ where O_2 is an orientation on G_{NCL} that differs from O_1 by the orientation of a single edge. Similarly, for an orientation O_3 on G_{NCL} obtained from O_1 by flipping the orientation of a single edge, there is a 1-switch operation that takes Π_{O_1} to Π_{O_3} .*

Proof. Consider an edge e in G whose endpoints are in different districts in Π_{O_1} . By construction of our gadgets and Lemma 2, $e = rt$ for an anchor r and a terminal t . Further, note that the district $\Pi_{O_1}(t)$ containing t cannot absorb r since the leaf adjacent to r would disconnect from the rest of its district, violating the requirement that districts stay connected. Thus, the only 1-switch we can perform along e is one in which the district $\Pi_{O_1}(r)$ containing r expands to t . This is precisely the district map Π_{O_2} where O_2 is the orientation we get by flipping the orientation of the edge associated with t in O_1 .

Conversely, let e' be the only edge in G_{NCL} whose orientation differs in O_1 and O_3 . By construction, e' corresponds to a terminal $v_{e'}$, which is adjacent to anchors in two distinct districts, and Π_{O_1} and Π_{O_3} differ only by the membership of $v_{e'}$. Hence Π_{O_3} is obtained from Π_{O_1} by performing a single 1-switch operation that moves $v_{e'}$ from one district to the other. \square

Lemma 19 implies the following theorem:

Theorem 20. *Given a graph G , and two k -district maps Π_A and Π_B on G , it is PSPACE-complete to determine whether Π_A and Π_B are in the same connected component of $\Gamma_k(G)$.*

4.2 PSPACE-Hardness for Planar Graphs

We now modify the reduction described in Section 4.1 so that it creates a planar graph G for a given NCL graph G_{NCL} . Recall that the NCL graph G_{NCL} is a planar cubic graph. Thus, to ensure that G is planar, it suffices that each gadget admits a planar drawing with terminals in the outer face.

The gadget associated with an AND vertex already satisfies this condition. In this section, we construct a slightly more complicated gadget which behaves like an OR vertex and admits a planar drawing with all of its terminals on the outer face. Refer to Figure 9. As before, the labeled vertices are the terminals for this gadget, and each terminal must be identified with the terminal of the neighboring gadget as previously described. Apart from terminals, there are three leafs and three anchors and an copy of K_3 (i.e., a 3-cycle) in the middle of the gadget. The three anchors are adjacent to distinct vertices of the 3-cycle.

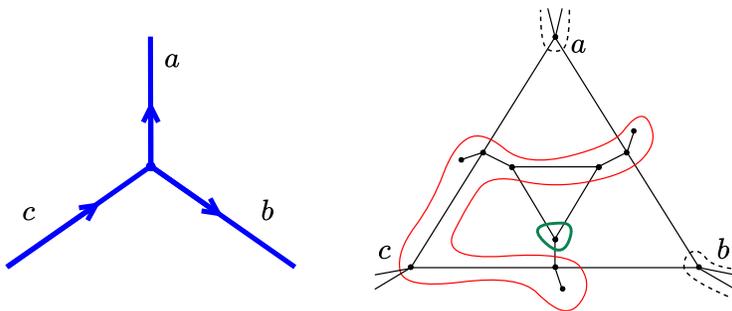


Figure 9: Modified planar OR gadget

Unlike the previous gadgets, this gadget comes equipped with two districts: one called the **guard** that interacts with other gadgets (shown in thin red in Figure 9), defined as the district that contains all three leafs and anchors of the gadget, and one called the **prisoner** that is trapped inside of this gadget (shown in bold green in Figure 9), defined as a district that is not the guard and that consists of some vertices of the 3-cycle. The correspondence between valid orientations of G_{NCL} , and district maps containing a guard and a unique prisoner district is defined as follows. At least one vertex in the 3-cycle in the gadget is in the prisoner district. We define only the prisoner district and let the guard district contain all remaining vertices of the gadget, except the terminals associated with edges of G_{NCL} oriented away from the corresponding vertex (see Figure 9). If the indegree of the OR vertex in G_{NCL} is 2 or 3, the prisoner district can be any nonempty set of vertices in the 3-cycle. Otherwise, the indegree of the OR vertex is exactly one, and the guard district will contain exactly one of the three terminals. In that case, the prisoner district contains either of the two vertices of the 3-cycle that are within distance 2 from the terminal vertex within the guard district. We now prove that, after any sequence of 1-switches, these properties of the guard and prisoner districts continue to hold, and they each induce a connected subgraph in G if and only if they correspond to a satisfying orientation of G_{NCL} .

Lemma 21. *The modified OR gadget behaves as the original OR gadget.*

Proof. As was the case for the original OR gadget, we can see that a satisfying orientation of the NCL graph will create a connected district map for this new gadget and vice versa (the guard district will be connected if and only if it contains at least one terminal, and the prisoner district is always connected because it consists of a subgraph of the complete graph K_3).

Once again, Lemma 2 guarantees that the three leaves and the adjacent anchors of a guard district remain in the same district under any sequence of switch operations. Further, since the prisoner is on a 3-cycle which is only adjacent to cut vertices (anchors) in the guard district, which remain in the same district by Lemma 2, the prisoner can never escape from this 3-cycle and the number of prisoners cannot change in a gadget. Thus, similarly to property (\star) in Section 4.1, we can uniquely identify an OR gadget with the two districts it must always contain.

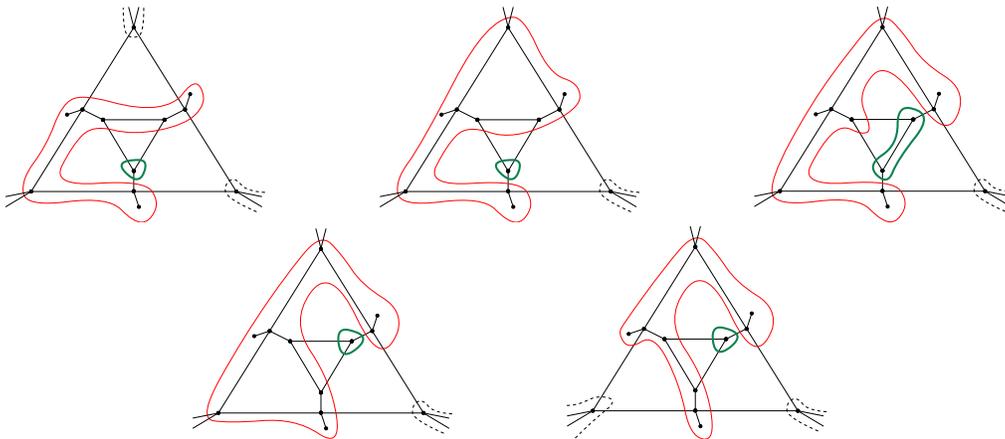


Figure 10: Transitioning from owning only the bottom left terminal to owning only the top terminal

We have seen that the modified OR gadget is satisfied in a static state by the same conditions as the original OR gadget; it remains to show that there is a valid transition between any two states corresponding to valid orientations of edges in G_{NCL} that differ by a flip.

As noted above, a guard district always contains all of its initial three anchors, and every terminal in its OR gadget is either in the guard district or adjacent to an anchor in the guard district. Therefore, a sequence of 1-switch operations can always transition to a state where the guard district contains two or more terminals (two or more incoming blue edges). If the guard gadget contains two or three terminals, the prisoner district can move freely within the central triangle. It follows that, from a state where the guard district contains two terminals (two incoming blue edges), the modified OR gadget can transition to a state where it contains only one of the original two (a single incoming blue edge). As an illustration, Figure 10 shows how the gadget can transition between states where the guard district contains any one single terminal (a single blue edge directed inwards) through intermediate states where it contains two (two blue edges oriented inwards). Note that there are two different district maps that correspond to the same orientation when there is a single blue edge is oriented towards an OR vertex.

Conversely, since the three terminals of a modified OR gadget are incident to distinct other gadgets (as G_{NCL} is a simple graph), a single switch can only add or remove one terminal to a guard district, hence only states that represent orientations differing by a single flip are adjacent in $\Gamma_k(G)$. \square

Theorem 22. *Given a planar graph G of maximum degree 6, and two k -district maps Π_A and Π_B over G , it is PSPACE-complete to determine if Π_A and Π_B are in the same connected component of $\Gamma_k(G)$.*

4.3 PSPACE-Hardness for Two-District Maps

In order to prove hardness in presence of only two districts, we modify the reduction described in Section 4.1 as follows. We start by subdividing every edge in the NCL graph G_{NCL} and creating degree-two vertices which are satisfied so long as they have in-degree at least one. The addition of these vertices has no effect on the reconfiguration space; these extra vertices simply propagate signals from one vertex to another. We can then assume that G_{NCL} is bipartite: one partite set containing only degree-2 vertices, and the other partite set containing the original AND and OR vertices.

To build a gadget that simulates degree-two NCL vertices, simply take the original nonplanar OR gadget and delete one of its terminals. The resulting gadget has two terminals, both of which provide independent paths between the two anchor/leaf pairs in the gadget, so the district corresponding to this gadget will be connected if and only if it owns at least one of its terminals. An example of this gadget appears in the center of Figure 11.

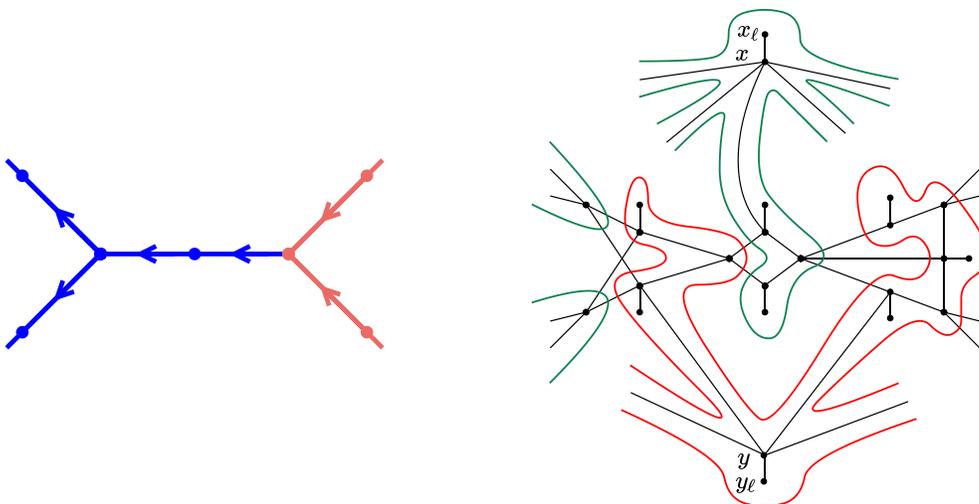


Figure 11: Transforming orientation on subdivided G_{NCL} to 2-district map over G .

Now, construct G from this subdivided version of G_{NCL} similar to Section 4.1 (using the original nonplanar OR gadget). The subdivided version of G_{NCL} has three types of vertices: OR, AND, and subdivision vertices. Each vertex is replaced by a gadget for the corresponding type. Next, create a vertex x with one leaf x_ℓ attached to it, and an edge connecting x to one anchor in every degree-two vertex gadget (either anchor is fine). Then create a vertex y with one leaf y_ℓ attached to it, and add an edge connecting y to one anchor in every OR and AND gadget (again any anchor is fine).

Finally, given an orientation on G_{NCL} , we start by building a district map on G in the same way as before, but after we have built this map, we merge all the districts on degree-two gadgets into a single district also containing x and x_ℓ , and then merge all the districts on AND and OR vertices into another single district also containing y and y_ℓ . This construction is shown in Figure 11.

Theorem 23. *Given a graph G and two 2-district maps Π_A and Π_B over G , it is PSPACE-complete to determine if Π_A and Π_B are in the same connected component of $\Gamma_2(G)$.*

Proof. Since the leaves x_ℓ and y_ℓ , and the leaves in every gadget must remain in their respective districts, and x and y are only adjacent to one anchor in each gadget, the connectivity requirements within each gadget remain, and thus every argument in Lemma 18 still applies. The two partite sets

of G_{NCL} form the basis of the two districts in our map. Thus, every adjacency between two gadgets in G is between gadgets in different districts, and conversely adjacency between the two districts is always between two neighboring gadgets. So all of the arguments in Lemma 19 still apply, yielding the stated result. \square

4.4 Exponential Diameter

The following theorem is implicit in the reduction from QSAT to NCL in [20].

Theorem 24. *For every $n \in \mathbb{N}$ there exist a planar NCL graph G_{NCL} on n vertices and initial and final orientations A, B on G_{NCL} such that $2^{\Theta(n)}$ edge flips are necessary and sufficient to reconfigure A into B .*

Proof. First, take the construction in Figure 4 of [20] showing an NCL graph which simulates a quantified formula evaluator. Now modify all n quantifier blocks in this construction to be universally quantified. Next, build any tautological boolean formula on n variables which can be expressed using a linear number of NCL gates; in particular the disjunction $x_1 \vee \bar{x}_1 \vee \dots \vee x_n \vee \bar{x}_n$ suffices. The resulting NCL graph thus has a total number of vertices and edges linear in the number of quantifiers. By Lemma 3 of the same paper we see that the i^{th} universal quantifier cannot have its satisfied-out edge flipped until the remainder of the quantified formula is evaluated under both variable assignments of x_i , so if a reconfiguration exists it requires at least $2^{\Omega(n)}$ edge flips; and in this case a reconfiguration is possible with $2^{O(n)}$ edge flips since all quantifiers are existential and the formula is a tautology. Since the original QSAT reduction contains some edge crossings, one might worry that in deploying crossover gadgets that ensure planarity we may see a quadratic blow-up in the size of the graph, weakening our bound to $2^{\Omega(\sqrt{n})}$ in the planar case. However, by inspection we see that each universal quantifier gadget contains only three edge crossings, and the formula $x_1 \vee \bar{x}_1 \vee \dots \vee x_n \vee \bar{x}_n$ can be constructed as a perfect binary tree with no crossings, so our lower bound holds in the planar case, as well. \square

Corollary 25. *The diameter of a connected component of $\Gamma_k(G)$ can be as large as $2^{\Omega(n)}$ where $n = |V(G)|$, even if G is a planar graph of maximum degree 6, or if $k = 2$.*

Proof. Since our reduction creates a redistricting instance over a graph linear in the size of the original NCL graph, this result follows from the reductions in Sections 4.1–4.3 and Theorem 24. \square

5 Hardness for Shortest Paths

In the previous section we showed that it is PSPACE-complete to decide whether two district maps on G are in the same component of $\Gamma_k(G)$. Hardness here crucially relied on the fact that $\Gamma_k(G)$ can have many connected components, each with potentially exponential diameter. In this section, we show that even if we constrain G to be biconnected, and thus $\Gamma_k(G)$ to be connected with polynomially bounded diameter (cf. Theorem 5), the problem of finding a shortest path from one district map to another in $\Gamma_k(G)$ is NP-hard. We start by showing NP-hardness for arbitrary graphs (Lemma 28) and then strengthen it to biconnected graphs (Theorem 29). The decision problem can be formally stated as follows: we are given a graph G , two k -district maps, Π_A and Π_B , and an integer $L \geq 0$, and ask whether a sequence of at most L switches can take Π_A to Π_B . Let us denote this problem by $R(G, \Pi_A, \Pi_B, L)$.

We present a polynomial-time reduction from 3SAT. An instance of 3SAT consists of a boolean formula φ in 3CNF. Let m and n be the number of clauses and the number of variables, respectively,

in φ . We construct, for a given 3SAT instance φ , a graph $G(\varphi)$, two district maps $\Pi_A(\varphi)$ and $\Pi_B(\varphi)$, and a nonnegative integer $L(\varphi)$. We then show that φ is satisfiable if and only if the instance $R(G(\varphi), \Pi_A(\varphi), \Pi_B(\varphi), L(\varphi))$ of the redistricting problem is positive.

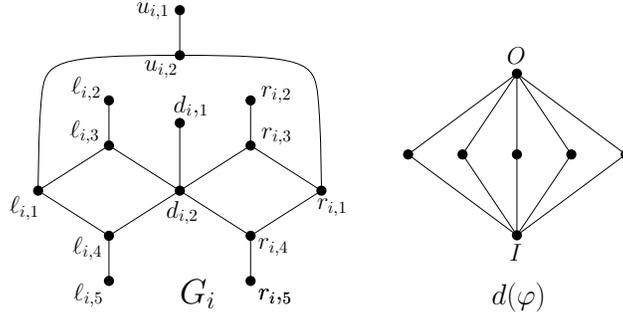


Figure 12: A variable gadget (left) and a districting pipe (right)

We construct the graph $G(\varphi)$ as follows:

1. For every variable x_i , construct a **variable gadget** G_i , shown in Figure 12 (left).
2. For every clause c_j , a **clause gadget** H_j consists of two adjacent vertices, $c_{j,1}$ and $c_{j,2}$. See Figure 13 (top).
3. For every variable x_i appearing in a clause c_j , if x_i is nonnegated (negated) in c_j , insert an edge between $c_{j,2}$ and $l_{i,1}$ ($r_{i,1}$).
4. Next, we add a subgraph, called a **districting pipe** $d(\varphi)$, that consists of $m+n+1$ vertices. The districting pipe is a complete bipartite graph between a 2-element partite set $\{O, I\}$ and a $(m+n-1)$ -element partite set. Figure 12 (right) depicts an example where $m+n-1=5$.
5. Lastly, for each variable gadget G_i , insert the edges $Ol_{i,1}$ and $Or_{i,1}$.

We now define two $(2m+5n+1)$ -district maps on $G(\varphi)$. Refer to Figure 13. First, let $\Pi_A(\varphi)$ consist of the following districts. For each variable gadget G_i , we create four districts: $l_i = \{l_{i,j} : j = 1, \dots, 5\}$, $r_i = \{r_{i,j} : j = 1, \dots, 5\}$, $\{d_{i,1}, d_{i,2}\}$, and $\{u_{i,1}, u_{i,2}\}$. For each clause gadget H_j , we create a 2-element district $\{c_{j,1}, c_{j,2}\}$. In the districting pipe, every vertex is in a singleton district, which yields $m+n+1$ singletons. Next, we define the target district map, $\Pi_B(\varphi)$. For every variable gadget G_i , we create similar districts to $\Pi_A(\varphi)$, the only difference is that the district $\{u_{i,1}, u_{i,2}\}$ is now split into two singletons: $\{u_{i,1}\}$ and $\{u_{i,2}\}$. In each clause gadget H_j , the two vertices form singleton districts. Lastly, the district pipe now consists of one $(m+n+1)$ -vertex district. Finally, we set $L(\varphi) := 4m+7n-1$. This completes the description of the instance $R(G(\varphi), \Pi_A(\varphi), \Pi_B(\varphi), L(\varphi))$.

Lemma 26. *If there exists a satisfying truth assignment for φ , then $R(G(\varphi), \Pi_A(\varphi), \Pi_B(\varphi), L(\varphi))$ is a positive instance.*

Proof. Let τ be a satisfying truth assignment for φ . We show that $\Pi_A(\varphi)$ can be transformed into $\Pi_B(\varphi)$ using $L(\varphi)$ switches. We define an open gate of G_i to be either $l_{i,1}$ if district l_i contains the vertex $d_{i,2}$, or $r_{i,1}$ if district r_i contains $d_{i,2}$. This implies that the open gate is a vertex in the variable gadget that can be taken by external districts. Note that l_i (resp., r_i) cannot leave vertices $l_{i,2}, l_{i,3}, l_{i,4}$, and $l_{i,5}$ (resp., $r_{i,2}, r_{i,3}, r_{i,4}$, and $r_{i,5}$) by Lemma 1. So then both $l_{i,1}$ and $r_{i,1}$ cannot be taken by external districts, or else l_i or r_i would be disconnected.

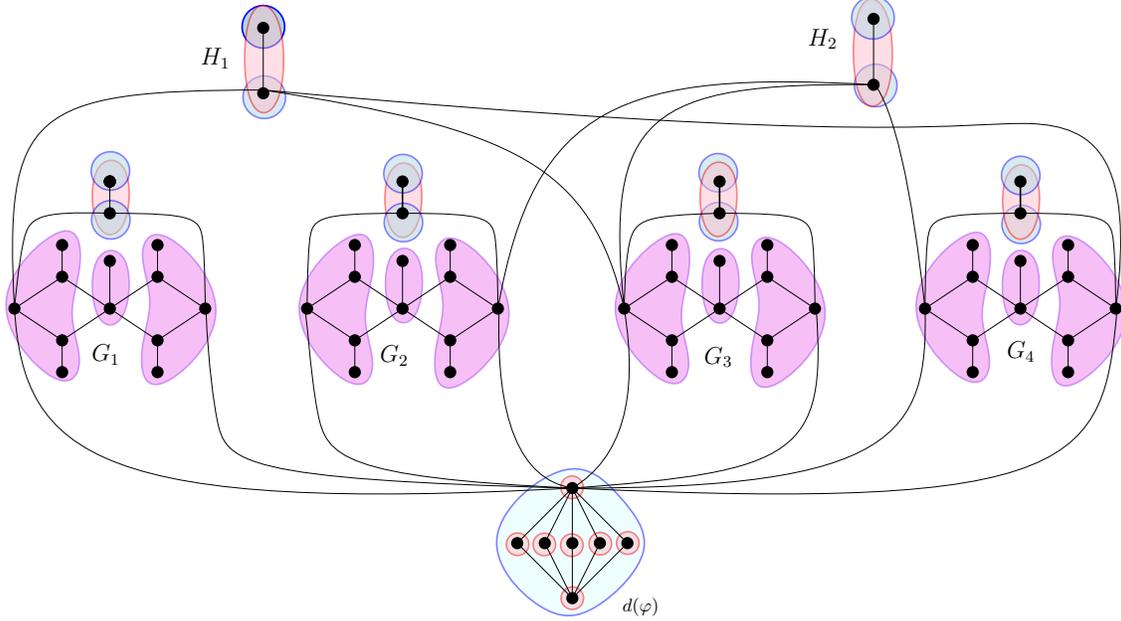


Figure 13: The graph $G(\varphi)$ for the formula $\varphi = (x_1 \vee x_3 \vee \neg x_4) \wedge (\neg x_2 \vee x_3 \vee x_4)$. The red and purple regions represent the districts in $\Pi_A(\varphi)$. The blue and purple regions represent the districts in $\Pi_B(\varphi)$. In particular, the purple regions represent districts that are present in both $\Pi_A(\varphi)$ and $\Pi_B(\varphi)$.

1. For each variable x_i , if $\tau(x_i) = \mathbf{true}$ (\mathbf{false}), then expand ℓ_i (r_i) into $d_{i,2}$ (thereby opening one of the two gates for each variable) in a total of n switches.
2. For every variable x_i , we transform the district containing O into a district containing only $u_{i,2}$ and the open gate of G_i by expanding twice (for $\tau(x_i) = \mathbf{true}$, $\mathbf{switch}(\ell_{i,3}, \ell_{i,1}, O)$ and $\mathbf{switch}(u_{i,1}, u_{i,2}, \ell_{i,1})$) and shrinking it out of O using either one, for x_1 , or two switches, for remaining variables (for $\tau(x_i) = \mathbf{true}$, $\mathbf{switch}(O, p_i, I)$ and $\mathbf{switch}(\ell_{i,1}, O, p_{i+1})$ where p_i and p_{i+1} are vertices in the districting pipe). We can perform this step in $3n + (n - 1)$ switches by, for the previously mentioned shrinks, expanding a singleton containing a degree-2 vertex of $d(\varphi)$ whenever possible, or expanding the district containing I .
3. For each clause c_j , choose any variable x_i that appears in a true literal in c_j (guaranteed to exist by the definition of τ). Transform the district containing O into a district containing only $c_{j,2}$ and the open gate of x_i , similar to the last step. Using the same strategy, this step can be accomplished with $4m$ switches. After this step $m + n$ districts were moved out of $d(\varphi)$, and now a single district contains all its vertices (the district initially containing I).
4. Finally, for every variable x_i we close its gate by first expanding either ℓ_i or r_i into the open gate and then expanding the singleton district at $d_{i,1}$ into $d_{i,2}$. This takes $2n$ switches.

Overall, we have performed $n + 3n + (n - 1) + 4m + 2n = 4m + 7n - 1 = L(\varphi)$ switches. These $L(\varphi)$ switches transformed $\Pi_A(\varphi)$ to $\Pi_B(\varphi)$, and so the instance $R(G(\varphi), \Pi_A(\varphi), \Pi_B(\varphi), L(\varphi))$ is positive. \square

Lemma 27. *If $R(G(\varphi), \Pi_A(\varphi), \Pi_B(\varphi), L(\varphi))$ is a positive instance of the redistricting problem for a boolean formula φ , then there exists a satisfying truth assignment for φ .*

Proof. We derive lower bounds on the number of switches in any sequence of switches from $\Pi_A(\varphi)$ to $\Pi_B(\varphi)$ by making inferences from the initial and target district maps. Notice that if a district contains a leaf, then the leaf remains in the same district by Lemma 1. We call a district **mobile** if it does not contain any leaf in $\Pi_A(\varphi)$. By construction, only the $m + n + 1$ districts initially in the districting pipe are mobile.

- (A) Since $u_{i,1}$ and $u_{i,2}$ are in distinct districts in $\Pi_B(\varphi)$, we must have a mobile district that travels to $u_{i,2}$. In order to accomplish this, we must first open one of the two gates of the variable gadget G_i . Opening n gates, one in each variable gadget, requires at least n switches.
- (B) As noted above, a mobile district must travel to $u_{i,2}$ for $i = 1, \dots, n$. Moving n mobile districts from O to $u_{i,2}$, $i = 1, \dots, n$, requires at least $2n$ switches, and an additional $2(n - 1)$ switches for $n - 1$ mobile districts to reach O : one switch expanding the district containing I and one expanding the desired mobile district to O . Overall, this requires at least $2n + 2(n - 1)$ switches.
- (C) Since each clause gadget H_j consists of two districts in $\Pi_B(\varphi)$, a mobile district from the districting pipe must travel to $c_{j,2}$, for $j = 1, \dots, m$, which requires $4m$ switches.
- (D) Because one mobile district must expand to the entire district pipe $d(\varphi)$, either one mobile district expands into I , or the district that contains I expands into O . In either case, this takes one additional move that has not been counted so far.
- (E) Note that the gate of G_i is closed and $\{d_{i,1}, d_{i,2}\}$ is a 2-vertex district in $\Pi_B(\varphi)$, for $i = 1, \dots, n$. So the district of the open gate must expand to consume its gate, and the singleton district at the leaf $d_{i,1}$ must expand into $d_{i,2}$. Together this requires a total of $2n$ switches.

Therefore, we need at least $n + 2n + 2(n - 1) + 4m + 1 + 2n = 7n + 4m - 1 = L(\varphi)$ switches to solve the redistricting problem. Since we executed exactly $L(\varphi)$ switches, no other move is allowed.

Due to the fact that after opening a gate, opening the opposite gate would require additional switches, we conclude that precisely one gate opens in each variable gadget. We construct a truth assignment as follows: For every $i = 1, \dots, n$, let $\tau(x_i) = \mathbf{true}$ if the left gate of the variable gadget G_i opens, and $\tau(x_i) = \mathbf{false}$ otherwise. Since the only way to get a district to $c_{j,2}$ was through an open gate of one of the three literal in the clause c_j , then every clause is incident to an open gate of a variable gadget. Since every open gate corresponds to a true literal, at least one of the three literals is true in each clause. Henceforth, τ is a satisfying truth assignment for φ . \square

The following Lemma is the direct consequence of Lemmas 26 and 27.

Lemma 28. *Given an graph G with n vertices and an integer $1 \leq k \leq n$, it is NP-complete to decide whether the length of a shortest path in $\Gamma_k(G)$ between two given district maps is below a given threshold.*

The unshrinkable districts in the previous reduction are not essential for NP-hardness. We can modify the reduction to produce a biconnected graph G as follows.

Theorem 29. *It is NP-hard to decide whether the length of a shortest path in $\Gamma_k(G)$ between two given shrinkable district maps is below a given threshold even when G is biconnected.*

Proof. In the reduction above, for a boolean formula φ in 3CNF, we constructed an instance $R(G(\varphi), \Pi_A(\varphi), \Pi_B(\varphi), L(\varphi))$ of the redistricting problem. We modify the reduction by subdividing the edge connected to every leaf creating a path of length $L(\varphi)$. Now connect every leaf to

the vertex I in $d(\varphi)$. The resulting graph is biconnected because the only cut vertices produced in the previous reductions were either O in $d(\varphi)$ or adjacent to leaves. The modification guarantees that they are no longer cut vertices. We make the following modifications to the district maps $\Pi_A(\varphi)$ and $\Pi_B(\varphi)$: If both endpoints of a subdivided edge belonged to the same district, add the new vertex created by the subdivision to this district; Extend the singleton districts that previously contained a leaf in $B(\varphi)$ to contain all new vertices on its path. The districts that contain long paths (of length $L(\varphi)$) cannot leave such paths completely since we are only allowed $L(\varphi)$ switches. Then, the only way to obtain $\Pi_B(\varphi)$ from $\Pi_A(\varphi)$ is to move the singleton districts in $d(\varphi)$ through the variable gadgets. The rest of the proof is analogous to the proof of Lemmas 26 and 27. \square

6 Conclusion

This paper provides the theoretical foundation for using elementary switch operations to explore the configuration space $\Gamma_k(G)$ of all partitions of a given graph into k nonempty subgraphs, each of which is connected. We gave a polynomial-time testable combinatorial characterization for connected configurations spaces (Theorem 17).

Our PSPACE-hardness proof with few (two) districts produces a nonplanar graph. The complexity of deciding whether two k -district maps (with $k = O(1)$) on a planar graph G are in the same component of $\Gamma_k(G)$ remains open. Our NP-hardness reduction for the shortest path problem produces a nonplanar (biconnected) graph. It is an open problem to determine the computational complexity of computing shortest paths in $\Gamma_k(G)$ when G is biconnected and planar, or in $\Gamma'_k(G)$ when G is planar.

A crucial concept in both the combinatorial characterization and the reconfiguration algorithms (Algorithms 1 and 2) was **shrinkability**: A district is shrinkable if it can be reduced to a single vertex (while all k districts remain connected). In applications to electoral maps, all districts have roughly average size, say between $\frac{n}{2k}$ and $\frac{2n}{k}$, and a singleton district is impractical. In a sense, we establish that there is a path between any two shrinkable district maps with average-size districts by passing through “impractical” district maps with singleton districts. We do not know whether singleton districts are necessary: for a constant $c \geq 1$, we can define $\Gamma_{k,c}(G)$ as the graph of k -district maps in which the size of every district lies in the interval $[\frac{n}{ck}, \frac{cn}{k}]$. It is easy to construct examples where $\Gamma_{k,1}(G)$ has isolated vertices. Is there a constant $c > 1$ such that the connectedness of $\Gamma_k(G)$ implies that $\Gamma_{k,c}(G)$ is also connected?

In our model, a district map is a partition of the vertex set into k **unlabeled** nonempty subsets. One could consider the **labeled** variant, and define a switch graph $\Gamma_k^L(G)$ on labeled k -district maps. Our results do not carry over to this variant: in particular, the labeled switch graph $\Gamma_k^L(G)$ need not be connected if G is biconnected. For example, if $G = C_n$ (i.e., a cycle of $n \geq 3$ vertices) and $k \geq 3$, then the cyclic order of the districts along the cycle cannot change. In the special case that $k = 2$ and G is biconnected, $\Gamma_2^L(G)$ is connected since we can shrink a district to a singleton (cf. Lemma 3) and move it to any vertex while the complement remains connected. When we move a singleton district from one vertex to another, it temporarily occupies both vertices, which should not form a 2-cut. Shrinking a district to a singleton is sometimes necessary in this case (one such example is $G = K_{2,m}$, $m \geq 3$, where the 2-element partite set is split between the two districts).

References

- [1] Nicola Apollonio, Ronald I. Becker, Isabella Lari, Federica Ricca, and Bruno Simeone. Bicolored graph partitioning, or: Gerrymandering at its worst. *Discrete Applied Mathematics*,

- 157(17):3601–3614, 2009. doi:10.1016/j.dam.2009.06.016.
- [2] Vincenzo Auletta, Angelo Monti, Mimmo Parente, and Pino Persiano. A linear-time algorithm for the feasibility of pebble motion on trees. *Algorithmica*, 23(3):223–245, 1999. doi:10.1007/PL00009259.
 - [3] Sachet Bangia, Christy Vaughn Graves, Gregory Herschlag, Han Sung Kang, Justin Luo, Jonathan C Mattingly, and Robert Ravier. Redistricting: Drawing the line. Preprint, arXiv:1704.03360, 2017. URL: <https://arxiv.org/abs/1704.03360>.
 - [4] Cedric Berenger, Peter Niebert, and Kévin Perrot. Balanced connected partitioning of unweighted grid graphs. In *Proc. 43rd Mathematical Foundations of Computer (MFCS)*, volume 117 of *LIPICs*, pages 39:1–39:18. Schloss Dagstuhl, 2018. doi:10.4230/LIPICs.MFCS.2018.39.
 - [5] Marthe Bonamy and Nicolas Bousquet. Token sliding on chordal graphs. In *43rd Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 10520 of *LNCS*, pages 127–139. Springer, 2017. doi:10.1007/978-3-319-68705-6_10.
 - [6] Édouard Bonnet, Tillmann Miltzow, and Paweł Rzażewski. Complexity of token swapping and its variants. *Algorithmica*, 80(9):2656–2682, 2018. doi:10.1007/s00453-017-0387-0.
 - [7] Janka Chlebíková. Approximating the maximally balanced connected partition problem in graphs. *Inf. Process. Lett.*, 60(5):223–230, 1996. doi:10.1016/S0020-0190(96)00175-5.
 - [8] Erik D. Demaine, Martin L. Demaine, Eli Fox-Epstein, Duc A. Hoang, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada. Linear-time algorithm for sliding tokens on trees. *Theor. Comput. Sci.*, 600:132–142, 2015. doi:10.1016/j.tcs.2015.07.037.
 - [9] Giuseppe Di Battista and Roberto Tamassia. On-line planarity testing. *SIAM J. Comput.*, 25(5):956–997, 1996. doi:10.1137/S0097539794280736.
 - [10] Moon Duchin. Gerrymandering metrics: How to measure? What’s the baseline? *Bulletin of the American Academy for Arts and Sciences*, 72(2):54–58, 2018. URL: <https://arxiv.org/abs/1801.02064>.
 - [11] Moon Duchin and Bridget Eileen Tenner. Discrete geometry for electoral geography. Preprint, arXiv:1808.05860, 2018. URL: <https://arxiv.org/abs/1808.05860>.
 - [12] Adrian Dumitrescu and János Pach. Pushing squares around. *Graphs and Combinatorics*, 22(1):37–50, 2006. doi:10.1007/s00373-005-0640-1.
 - [13] Martin E. Dyer and Alan M. Frieze. On the complexity of partitioning graphs into connected subgraphs. *Discrete Appl. Math.*, 10(2):139–153, 1985. doi:10.1016/0166-218X(85)90008-3.
 - [14] Benjamin Fifield, Michael Higgins, Kosuke Imai, and Alexander Tarr. Automated redistricting simulation using Markov chain Monte Carlo. *Journal of Computational and Graphical Statistics*, 29(4):715–728, 2020. doi:10.1080/10618600.2020.1739532.
 - [15] Balázs Fleiner, Balázs Nagy, and Attila Tasnádi. Optimal partisan districting on planar geographies. *CEJOR*, 25(4):879–888, 2017. doi:10.1007/s10100-016-0454-7.

- [16] Klaus-Tycho Foerster, Linus Groner, Torsten Hoefler, Michael König, Sascha Schmid, and Roger Wattenhofer. Multi-agent pathfinding with n agents on graphs with n vertices: Combinatorial classification and tight algorithmic bounds. In *Proc. 10th Conf. Algorithms and Complexity (CIAC)*, volume 10236 of *LNCS*, pages 247–259. Springer, 2017. doi:10.1007/978-3-319-57586-5_21.
- [17] Shinya Fujita, Tomoki Nakamigawa, and Tadashi Sakuma. Colored pebble motion on graphs. *Eur. J. Comb.*, 33(5):884–892, 2012. doi:10.1016/j.ejc.2011.09.019.
- [18] Oded Goldreich. Finding the shortest move-sequence in the graph-generalized 15-puzzle is NP-hard. In *Studies in Complexity and Cryptography*, volume 6650 of *LNCS*, pages 1–5. Springer, 2011. doi:10.1007/978-3-642-22670-0_1.
- [19] Gilad Goralý and Refael Hassin. Multi-color pebble motion on graphs. *Algorithmica*, 58(3):610–636, 2010. doi:10.1007/s00453-009-9290-7.
- [20] Robert A. Hearn and Erik D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoretical Computer Science*, 343(1):72–96, 2005. doi:10.1016/j.tcs.2005.05.008.
- [21] Gregory Herschlag, Robert Ravier, and Jonathan C. Mattingly. Evaluating partisan gerrymandering in Wisconsin. Preprint, 2017. URL: <https://arxiv.org/abs/1709.01596>.
- [22] John E. Hopcroft and Robert E. Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, 1973. doi:10.1145/362248.362272.
- [23] Anvesh Komuravelli, Arnab Sinha, and Arijit Bishnu. Connectivity preserving transformations for higher dimensional binary images. *Discrete Applied Mathematics*, 157(16):3372–3385, 2009. doi:10.1016/j.dam.2009.03.011.
- [24] Daniel Kornhauser, Gary L. Miller, and Paul G. Spirakis. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications. In *25th IEEE Symp. Foundations of Computer Science (FOCS)*, pages 241–250, 1984. doi:10.1109/SFCS.1984.715921.
- [25] Richard Kueng, Dustin G. Mixon, and Soledad Villar. Fair redistricting is hard. *Theoretical Computer Science*, 791:28–35, 2019. doi:10.1016/j.tcs.2019.04.004.
- [26] J. Leños and A. L. Trujillo-Negrete. The connectivity of token graphs. *Graphs and Combinatorics*, 34(4):777–790, 2018. doi:10.1007/s00373-018-1913-9.
- [27] Tillmann Miltzow, Lothar Narins, Yoshio Okamoto, Günter Rote, Antonis Thomas, and Takeaki Uno. Approximation and hardness of token swapping. In *24th European Symposium on Algorithms (ESA)*, volume 57 of *LIPIcs*, pages 66:1–66:15. Schloss Dagstuhl, 2016. doi:10.4230/LIPIcs.ESA.2016.66.
- [28] Ruy Fabila Monroy, David Flores-Peñaloza, Clemens Huemer, Ferran Hurtado, Jorge Urrutia, and David R. Wood. Token graphs. *Graphs and Combinatorics*, 28(3):365–380, 2012. doi:10.1007/s00373-011-1055-9.
- [29] Clemens Puppe and Attila Tasnádi. A computational approach to unbiased districting. *Mathematical & Computer Modelling*, 48(9-10):1455–1460, 2008. doi:10.1016/j.mcm.2008.05.024.

- [30] Clemens Puppe and Attila Tasnádi. Optimal redistricting under geographical constraints: Why “pack and crack” does not work. *Economics Letters*, 105(1):93–96, 2009. doi:10.1016/j.econlet.2009.06.008.
- [31] Daniel Ratner and Manfred K. Warmuth. $N \times N$ puzzle and related relocation problem. *J. Symb. Comput.*, 10(2):111–138, 1990. doi:10.1016/S0747-7171(08)80001-6.
- [32] Federica Ricca, Andrea Scozzari, and Bruno Simeone. Weighted Voronoi region algorithms for political districting. *Mathematical and Computer Modelling*, 48(9-10):1468–1477, 2008. doi:10.1016/j.mcm.2008.05.041.
- [33] Federica Ricca, Andrea Scozzari, and Bruno Simeone. Political districting: from classical models to recent approaches. *Annals OR*, 204(1):271–299, 2013. doi:10.1007/s10479-012-1267-2.
- [34] Federica Ricca and Bruno Simeone. Local search algorithms for political districting. *European J. Operational Research*, 189(3):1409–1426, 2008. doi:10.1016/j.ejor.2006.08.065.
- [35] Peter Sanders and Christian Schulz. Distributed evolutionary graph partitioning. In *Proc. 14th Meeting on Algorithm Engineering & Experiments (ALENEX)*, pages 16–29. SIAM, 2012. doi:10.1137/1.9781611972924.2.
- [36] Christian Schulz. Graph partitioning and graph clustering in theory and practice. Lecture notes, 2016. URL: <http://algo2.iti.kit.edu/2577.php>.
- [37] Richard M. Wilson. Graph puzzles, homotopy, and the alternating group. *Journal of Combinatorial Theory, Series B*, 16(1):86–96, 1974. doi:10.1016/0095-8956(74)90098-7.
- [38] Katsuhisa Yamanaka, Erik D. Demaine, Takehiro Ito, Jun Kawahara, Masashi Kiyomi, Yoshio Okamoto, Toshiki Saitoh, Akira Suzuki, Kei Uchizawa, and Takeaki Uno. Swapping labeled tokens on graphs. *Theor. Comput. Sci.*, 586:81–94, 2015. doi:10.1016/j.tcs.2015.01.052.
- [39] Katsuhisa Yamanaka, Takashi Horiyama, J. Mark Keil, David G. Kirkpatrick, Yota Otachi, Toshiki Saitoh, Ryuhei Uehara, and Yushi Uno. Swapping colored tokens on graphs. *Theor. Comput. Sci.*, 729:1–10, 2018. doi:10.1016/j.tcs.2018.03.016.