

An FPT Algorithm for Diverse Minimum s – t Cuts

Krishnan Dehaleesan 

University of Bergen, Norway

Pål Grønås Drange 

University of Bergen, Norway

Fedor V. Fomin 

University of Bergen, Norway

Petr A. Golovach 

University of Bergen, Norway

Laure Morelle 

University of Bergen, Norway

Abstract

We study the problem of finding a family of diverse minimum edge s – t cuts in a directed weighted graph G . Given integers k and d , the task is to decide whether G contains k minimum s – t cuts $C_1, \dots, C_k$ such that for any $i, j \in [k]$, the number of edges in the symmetric difference $C_i \Delta C_j$ is at least d .

For $d \in \{1, 2\}$, the problem corresponds to counting minimum s – t cuts in G , which is $\#P$ -complete [Provan and Ball, SICOMP 1983]. The problem is also known to be NP-complete already for $k = 3$ [de Berg, López Martínez, Spieksma, ISAAC 2024]. Our main result shows that the problem is fixed-parameter tractable (FPT) when parameterized by the combined parameter $k + d$.

The main ingredients of our FPT algorithm build on novel structural properties of diverse minimum s – t cuts and a non-trivial application of the flow-augmentation technique of Kim, Kratsch, Pilipczuk, and Wahlström [JACM 2025].

2012 ACM Subject Classification Theory of computation $\rightarrow$ Parameterized complexity and exact algorithms; Theory of computation $\rightarrow$ Graph algorithms analysis; Theory of computation $\rightarrow$ Network flows

Keywords and phrases Minimum s – t -cut, diversity, parameterized complexity, color coding, flow-augmentation

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.148

Related Version *Full Version:* <https://arxiv.org/abs/2607.03266> [9]

Funding *Fedor V. Fomin:* Supported by the Research Council of Norway under BWCA project (grant no. 314528) and by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (NewPC grant agreement No. 101199930)

Petr A. Golovach: Supported by the Research Council of Norway under the BWCA (grant no. 314528) and Extreme-Algorithms (grant no 355137) projects.

Laure Morelle: Supported by the Research Council of Norway under the Extreme-Algorithms (grant no 355137) project.

1 Introduction

The MINIMUM s – t CUT problem is one of the most fundamental and central problems in graph algorithms. Given a directed graph with designated vertices s and t , the task is to separate s from t by removing a set of edges of minimum total capacity. This problem is deeply connected to the theory of network flows, as formalized by the max-flow min-cut theorem, and to notions of graph connectivity.



© Krishnan Dehaleesan, Pål Grønås Drange, Fedor V. Fomin, Petr A. Golovach, and Laure Morelle; licensed under Creative Commons License CC-BY 4.0

34th Annual European Symposium on Algorithms (ESA 2026).

Editors: Philip Bille, Seth Pettie, and Sabine Storandt; Article No. 148; pp. 148:1–148:17

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

It is well known that the MINIMUM s - t CUT problem is solvable in polynomial time via the connection to the MAXIMUM FLOW problem, and a long line of research has culminated in an almost-linear time $O(m^{1+o(1)})$ [5]. However, several natural variants quickly become computationally more challenging. In particular, counting the number of minimum s - t cuts is #P-hard [22]. On the other hand, the maximum number of *disjoint* minimum s - t cuts could be easily found in polynomial time [26]. (One proceeds iteratively by computing the leftmost minimum cut and contracting its edges.)

In applications, one often wants not just a single minimum s - t cut, but several optimal alternatives from which a user can choose. A naive approach may return cuts that are almost identical, differing in only a few edges. It is therefore natural to ask for a family of minimum s - t cuts that is *diverse*, meaning that every pair differs in at least $d > 0$ edges. This perspective fits into the broader study of diversity in combinatorial optimization [4, 12, 13, 16, 8]. In this context, de Berg et al. [7] introduced the notion of *diverse* minimum s - t cuts. The problem of identifying diverse minimum s - t cuts occupies an “intermediate” position between counting all minimum cuts (#P-complete) and computing pairwise disjoint minimum cuts (in P).

For a family of minimum s - t cuts $C_1, C_2, \dots, C_k \subseteq E(G)$ in a graph G , de Berg et al. define the following natural notions of their diversity:

Name	Objective	Interpretation
min	$\min_{i,j} C_i \Delta C_j $	Maximize the minimum pairwise “disjointness”.
cov	$ \bigcup_i C_i $	Maximize coverage of the solution space.
sum	$\sum_{i,j} C_i \Delta C_j $	Maximize the total pairwise “disjointness”.

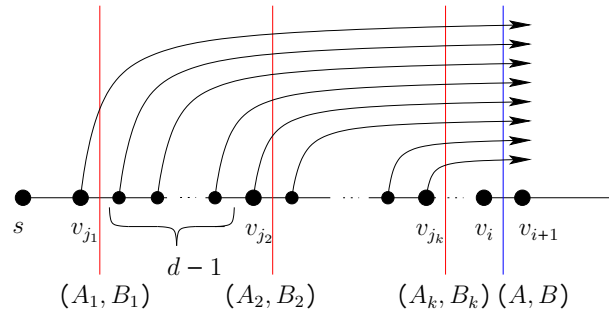
Here $C_i \Delta C_j$ denotes the symmetric difference of the edge sets of cuts C_i and C_j .

By making use of submodular function minimization, de Berg et al. [7] provided a (strongly) polynomial-time algorithm for computing minimum s - t cuts with respect to the *cov* and *sum* measures of diversity. Interestingly, the computational complexity of the *min* diversity measure appears to be more elusive. As shown by de Berg et al. [7], the problem of finding $k = 3$ minimum s - t cuts under the *min* diversity measure is already NP-hard. Meanwhile, when $k = 2$, the three measures converge, and the solution is always to take the two extremal cuts, i.e., the minimum s - t cut closest to s , and the minimum s - t cut closest to t [7]. Our main results shed light on the parameterized complexity of this intriguing measure of diversity.

1.1 Overview of Our Results

We consider a more general capacitated (weighted) variant of the diverse minimum s - t cuts problem defined by de Berg et al. [7]. While the weight of a cut is measured by the total capacity of its edges, the diversity between cuts is not measured by capacities, but rather by the number of edges in the symmetric difference of their edge sets. Because of that, it is convenient to formulate the problem using two cost functions on the edges of the input graph. The first cost function c , which we call the *capacity*, represents the weight of an edge and contributes to the total weight of a cut. The second cost function, called the *multiplicity*, arises from the algorithmic process.

In our algorithm, we contract certain edges of the graph, which may result in a graph containing parallel edges. Since we subsequently delete parallel edges, it is necessary to preserve their potential contribution both to cut weights and to the diversity measure. To maintain the correct contribution of a set of parallel edges to a cut, we update the capacity



■ **Figure 1** Construction of k d -diverse cuts; the cuts (A, B) and $(A_1, B_1), \dots, (A_k, B_k)$ are shown by vertical lines. Note that it may happen that $s = v_{j_1}$ or $v_{j_k} = v_i$.

of the resulting edge accordingly. However, since the diversity of cuts is defined via the symmetric difference of their edge sets, we must additionally maintain the multiplicity ℓ of edges.

Here is the formal definition of the problem.

DIVERSE MIN s - t -CUTS

Input: A flow network $\mathbb{G} = (G = (V, E), c : E(G) \rightarrow \mathbb{Z}_{\geq 1}, \ell : E(G) \rightarrow \mathbb{Z}_{\geq 1}, s, t)$, where c is the *capacities* and ℓ the *multiplicities* of edges, and s and t are two terminals, together with positive integers d and k .

Task: Decide whether there exists a family of k minimum capacity s - t cuts $C_1, \dots, C_k \subseteq E$ such that, for each $i, j \in [k]$, $\ell(C_i \triangle C_j) \geq d$.

With unit capacities and multiplicities, this corresponds to the *min* measure studied by de Berg et al. [7].

For $d \in \{1, 2\}$, this is the problem of finding k distinct minimum cuts, but when d is part of the input, the problem is NP-complete, even for $k = 3$ [7]. For unweighted graphs (graphs with unit capacities and multiplicities), by putting d to be twice the size of the minimum cut, we obtain the problem of computing of k pairwise disjoint minimum cuts, which is solvable in polynomial time [7, 26].

1.2 Outline of the Algorithm

In our main result, we show that the problem is FPT when parameterized by k and d .

► **Theorem 1.** *DIVERSE MIN s - t -CUTS can be solved in time $(kd)^{\mathcal{O}(k^s d^8)} \cdot n^{\mathcal{O}(1)}$.*

The algorithm for Theorem 1 is constructive. If the input graph contains k d -diverse minimum s - t cuts, the algorithm finds these cuts within the same running time.

The algorithm is obtained by combining novel structural properties of diverse minimum s - t cuts and the flow-augmentation technique of Kim et al. [18]. In the first step of the algorithm, we preprocess the input in polynomial time and either solve the problem or obtain an equivalent instance, where the number of edges in all minimum s - t cuts is bounded. The second step of the algorithm solves the problem when the number of edges in the minimum cut is bounded.

Lemmas 6 and 18 together provide the proof of Theorem 1. Indeed, Lemma 6 promises to find k d -diverse cuts or output an equivalent instance in which all the minimum cuts have a size bounded by $(kd)^2$. In the latter case, we make use of Lemma 18 that guarantees an FPT algorithm when the size of the minimum s - t cuts are bounded. Substituting the bound $r = (kd)^2$, Theorem 1 follows.

We briefly sketch the main ideas of the proof of Lemma 6 for the case when initially, both the capacity and multiplicity functions have unit values. The actual proof for the general case hinges on the same ideas but is more technical.

Given a directed graph G with a source and sink vertices s and t , respectively, we identify the *critical* edges, that is, edges that are included in some minimum s - t cut. Then we preprocess the instance by contracting some *residual*, i.e. non-critical, edges. By contractions, we obtain a new instance where the capacities and multiplicities get modified. If an edge-contraction creates multiple edges, we replace them by a single edge whose capacity and multiplicity is the sum of these values of the multiple edges.

We want to contract edges in such a way that minimum s - t cuts are not changed. While for undirected graphs, contracting all residual edges does not change minimum s - t cuts, this is not true for directed graphs. Here we have to be more careful. We use classical Menger's theorem (see, e.g., [10, 3]) that G has an s - t cut of size μ if and only if G has μ edge-disjoint s - t paths $P_1, \dots, P_\mu$. First, we contract residual edges of these paths. Then we contract the edges of all strongly connected components of $G - \bigcup_{i=1}^{\mu} E(P_i)$. Finding diverse minimum s - t cuts in the reduced instance is equivalent to finding diverse minimum s - t cuts in the original one.

The reduced graph G^* has a very particular structure. If we reverse the directions of all residual edges of G^* , we obtain a directed acyclic graph $\tilde{G}$. An interesting, and somewhat surprising, property of $\tilde{G}$ is that for every minimum cut $C = E(A, B)$ of G^* , there exists a topological ordering $v_1, \dots, v_n$ of $\tilde{G}$ such that $A = \{v_1, \dots, v_i\}$ for some $i \in [n]$. Moreover, for each $i \in [n]$, the partition (A, B) with $A = \{v_1, \dots, v_i\}$ and $B = \{v_{i+1}, \dots, v_n\}$ defines a minimum cut in G^* .

By leveraging these properties of cuts in G^* , we can conclude that if the preprocessed instance admits a minimum cut with at least $(kd)^2$ edges, then the graph contains k minimum s - t cuts that are pairwise d -diverse. Indeed, since a minimum cut $C = E(A, B)$ with $(kd)^2$ edges can be expressed as $A = \{v_1, \dots, v_i\}$ and $B = \{v_{i+1}, \dots, v_n\}$ for some topological ordering of $\tilde{G}$, at least $d(k-1) + 1$ endpoints of the cut edges must lie in either A or B . Indeed, otherwise with $d(k-1)$ points on either side, the maximum number of edges from A to B is limited to $d^2(k-1)^2 \leq (kd)^2$.

Without loss of generality, suppose that A contains at least $d(k-1) + 1$ such endpoints. Since for every $j \leq i$, the partition $A' = \{v_1, \dots, v_j\}$ and $B' = \{v_{j+1}, \dots, v_n\}$ also defines a minimum cut, we can select k minimum cuts whose pairwise symmetric differences are of size at least d .

The proof of Lemma 6 is constructive and allows us to find the actual cuts. Because a minimum s - t cut with maximum number of edges can be found in polynomial time, the problem is either solved or is reduced to the case when every minimum s - t cut contains less than $(kd)^2$ edges.

The algorithm in the proof of Lemma 18 consists of three major steps. First, we use the *color coding* technique of Alon, Yuster, and Zwick [2] to highlight edges in different cuts. The aim here is to find each of the cuts separately. For this, notice that the edges of $\bigcup_{i=1}^k C_i$ for a hypothetical solution $C_1, \dots, C_k$ can be partitioned into $2^k - 1$ classes according to their inclusion to cuts, and at most rk classes are nonempty because each cut contains at most r edges. Hence, we can guess nonempty classes and highlight their edges. By this step, the problem boils down to finding a minimum s - t cut with specific constraints on the colors of its edges and their multiplicities.

In the second step, we find such cuts using the flow augmentation technique. The main result of Kim et al. [18] (see Theorem 22) is that it is possible to augment a directed graph by adding some edges in such a way that an arbitrary minimal s - t cut becomes minimum.

Notice that we are looking for cuts of size at most r in the (non-capacitated) graph G that are inclusion-minimal. Thus, by making use the flow-augmentation technique, we further reduce the problem to the case when we have to find a certain minimum s - t cut which remains minimum if we forget about the edge capacities.

Finally, to solve the obtained problem, we again use Menger's theorem. We find a maximum family of at most r edge-disjoint s - t paths in G and make guesses about the colors and multiplicities of the edges from a solution cut in each path. After that, we can check whether there is a minimum s - t cut satisfying the given constraints using the standard minimum s - t cut algorithm.

1.3 Related work

The structure of minimum cuts has been studied extensively, beginning with the classical work of Picard and Queyranne, who characterized the full family of minimum cuts in a network and established their lattice structure [21]. From an algorithmic perspective, Karger and Stein gave a randomized algorithm for computing global minimum cuts [17]. Wagner studied a related but distinct notion of diversity by considering the problem of finding k pairwise-disjoint s - t -cuts of minimum total cost in edge-weighted graphs [26]. More broadly, diversity has appeared in optimization through the maximum diversity problem, which was analyzed and modeled via zero-one programming by Kuo et al. [20].

Recent work has explored diversity across many combinatorial problems, including minimum cuts [7], matchings [12], vertex cover [4], shortest paths [15], spanning trees [16], and travelling salesman [8]. Most relevant to our setting, Hanaka et al. recently showed that finding k pairwise-disjoint global minimum cuts is NP-hard [14], sharply contrasting with the tractability of classical minimum cut problems and motivating the study of alternative notions of diversity among minimum cuts.

Specifically, as mentioned above, de Berg et al. [7] consider all three diversity measures for DIVERSE MIN s - t -CUTS, proving that the problem is solvable in polynomial-time with respect to the *cov* and the *sum* measure, but NP-hard with respect to the *min* measure. They also provide a polynomial-time algorithm for computing disjoint minimum s - t cuts.

Drabik and Masařík show that diverse solutions for a wide class of vertex problems can be computed in FPT time parameterized by cliquewidth by extending dynamic programs on cliquewidth decompositions to support general diversity measures [11].

2 Preliminaries

We consider directed graphs with neither loops nor multiple edges. The vertex set of a graph G is denoted by $V(G)$ and the edge set by $E(G)$. We set $n = |V(G)|$ and $m = |E(G)|$. We refer the reader to [1] for the definition of *flow* and related terminology. We assume the reader is familiar with basic parameterized complexity, but for terminology and concepts on parameterized algorithms and complexity not defined in this article, we refer the reader to the textbook by Cygan et al. [6]. Throughout the paper, $c : E \rightarrow \mathbb{Z}_{\geq 1}$ is the edge *capacity* and $\ell : E \rightarrow \mathbb{Z}_{\geq 1}$ is the edge *multiplicity* (which can alternatively be seen as the number of parallel edges). For a set $X \subseteq E(G)$, we extend ℓ additively by defining $\ell(X) = \sum_{e \in X} \ell(e)$. A *flow network* is a tuple $\mathbb{G} = (G, c, \ell, s, t)$, where s is the source and t the target. We will refer directly to $G = (V, E)$ in the flow network when it is clear from context. For $a < b \in \mathbb{Z}$, we define $[a, b] = \{a, a + 1, \dots, b\}$ and $[b] = [1, b]$.

► **Definition 2** (Minimum s - t cut). Let $\mathbb{G}$ be a flow network and $d, k \in \mathbb{Z}_{\geq 1}$. An s - t cut, denoted (A, B) , is a partitioning of the vertices V into $A, B = V \setminus A$ such that $s \in A$ and $t \in B$. We will denote by $E(A, B)$ the set of edges going from A to B . Slightly abusing notation, we also refer to the edge set $C = E(A, B)$ as the cut. The capacity of a cut is

$$c(A, B) = c(E(A, B)) = \sum_{e \in E(A, B)} c(e).$$

The cut (A, B) is a minimum s - t cut if it is a cut of minimum capacity, whose value we denote by $\mu(G)$. Equivalently, a s - t -cut is minimum if its capacity is equal to the maximum flow value.

► **Definition 3** (d -diverse cut). Two cuts C, C' are said to be d -diverse if $\ell(C \Delta C') \geq d$, where Δ denotes symmetric difference. The cuts $C_1, \dots, C_k$ are said to be d -diverse if they are pairwise d -diverse.

Conventions. Note that every edge has a capacity and multiplicity of at least one. We will assume that the network has a non-zero maximum flow, i.e., there is a path from s to t . We also assume that s is a source-vertex and that t is a sink-vertex, i.e., $\deg^-(s) = 0$ and $\deg^+(t) = 0$. When we say minimum cuts, we refer to minimum s - t cuts unless stated otherwise. We start by recalling some preliminary properties of minimum s - t cuts.

► **Lemma 4** (Flow-value Lemma [19]). For any s - t -cut (A, B) , and flow f , the value of the total flow

$$v(f) = \sum_{e \in E(A, B)} f(e) - \sum_{e \in E(B, A)} f(e).$$

► **Lemma 5.** Let $\mu(G)$ be the size of a minimum s - t edge cut in a graph G with unit capacities. Then, it is possible to compute $\mu(G)$ edge-disjoint s - t paths in $\mathcal{O}(\min\{m^{1/2}, n^{2/3}\}m)$ time.

Proof. This follows directly from the edge version of Menger's theorem (see, e.g., [3, 10]). Alternatively, using Dinic's algorithm [1], the maximum s - t flow in a directed unit-capacity network can be computed deterministically in $\mathcal{O}(\min\{m^{1/2}, n^{2/3}\}m)$ time.

Extracting the paths: the max flow is integral (0 or 1 on each edge). Repeatedly start at s , follow edges with flow 1 until you reach t ; record that path and remove the edges in it and repeat until no such path remains. This takes $\mathcal{O}(m)$ total over all paths. ◀

3 Preprocessing

In this section, we prove the following important lemma, that we can preprocess our input graph in such a way that it has at most $(kd)^2$ edges.

► **Lemma 6.** There is an algorithm that, given an instance $(\mathbb{G}, d, k)$ of DIVERSE MIN s - t -CUTS, outputs in polynomial time either:

- that $(\mathbb{G}, d, k)$ is a yes-instance of DIVERSE MIN s - t -CUTS, or
- an equivalent instance $(\mathbb{G}^*, d, k)$ of DIVERSE MIN s - t -CUTS such that any minimum s - t cut in G^* has at most $(kd)^2$ edges.

3.1 Preliminary Results

Let us first prove some small preliminary results. First, we recall the folklore result that the minimum s - t cut with maximum number of edges can be found in polynomial time; we provide the proof for completeness.

► **Lemma 7.** *There is an algorithm that, given a directed graph G , a capacity $c : E(G) \rightarrow \mathbb{Z}_{\geq 1}$, and terminals $s, t \in V(G)$, outputs a minimum s - t cut with the maximum number of edges in polynomial time.*

Proof. Let $M = m + 1$. Let $c' : E(G) \rightarrow \mathbb{Z}_{\geq 1}$ be the capacity such that $c'(e) = c(e) \cdot M - 1$ for each edge e . For a cut (A, B) , its capacity with respect to c' is

$$\sum_{e \in E(A, B)} c'(e) = M \cdot \sum_{e \in E(A, B)} c(e) - |E(A, B)|.$$

Thus, a minimum s - t cut of G with respect to c' is a minimum s - t cut of G with respect to c that maximize its number of edges. Thus, it suffices to compute minimum s - t cut of G with respect to c' , which can be done in polynomial time [5]. ◀

We observe that edge contractions preserve flows. Let G be a graph with given source and sink vertices s and t , respectively, and a capacity function $c : E(G) \rightarrow \mathbb{Z}_{\geq 1}$. Let also f be a feasible s - t flow. Consider an arbitrary edge $(u, v) \neq (s, t)$. As $f : E(G) \rightarrow \mathbb{Z}_{\geq 0}$, we can use the same convention as with other weight functions, and assume that if the contraction of (u, v) creates multiple edges, then we define the flow on the resulting single edge by taking the sum of the flows on the multiple edges. We say that the obtained function is an *induced flow*.

► **Lemma 8.** *Let $G = (V, E)$ be a directed s, t flow network with capacities, and let f be a feasible s - t flow in G . Let $e = (u, v) \in E$ be an edge distinct from (s, t) . Let G' be the graph obtained from G by contracting e , and let f' the induced flow. Then f' is a valid s - t flow and $v(f') = v(f)$.*

Proof. Notice that the flow conservation at the single vertex obtained from u and v is preserved. Thus, f' is a flow. As $(u, v) \neq (s, t)$, $v(f') = v(f)$. ◀

Let G be a directed graph with edge capacity c and let $s, t \in V(G)$ be terminals. Let C be the set of all edges that appear in some minimum s - t cut, and $R = E \setminus C$. We call C to be the set of *critical edges* of the graph and R to be the set of *residual edges* of the graph. Let $G_R = (V(G), R)$ represent the residual graph.

By the well-known properties of minimum s - t cuts and maximum flows, we can make the following observation.

► **Observation 9.** *If an edge is critical, then it is saturated in every maximum flow. Furthermore, an edge is critical if and only if its deletion reduces the maximum flow by the capacity of e .*

Recall also the following property of minimum s - t cuts and maximum flows.

► **Observation 10.** *Let (A, B) be a minimum s - t cut in a graph G and let f be a maximum s - t flow. Then every edge $(u, v) \in E(B, A)$ carries zero flow.*

Proof. Since f is a maximum flow and (A, B) is a minimum s - t cut, we have $v(f) = c(A, B)$. It follows from Observation 9 (since $(u, v) \in E(A, B)$ edges are critical) that

$$\sum_{u \in A, v \in B} f(u, v) = c(A, B) \quad \text{and} \quad \sum_{u \in B, v \in A} f(u, v) = 0.$$

This means that every edge from B to A carries zero flow. $\blacktriangleleft$

Finally, we observe that strongly connected components of the residual graph are not separated by minimum s - t cuts.

► **Lemma 11.** *Let (S, T) be a minimum s - t cut. Let $G_{R'} = (V', R')$ be a strongly connected component of G_R . Then $V' \subseteq S$ or $V' \subseteq T$.*

Proof. Assume toward a contradiction that V' has vertices in S and T . This means that we have edges e and f that go from S to T and T to S respectively, since $G_{R'}$ is a strongly connected component. However, this means that e belongs to the minimum s - t cut induced by (S, T) which contradicts the fact that e is a residual edge. $\blacktriangleleft$

3.2 Reducing Large Cuts

Let $(\mathbb{G}, d, k)$ be an instance of DIVERSE MIN s - t -CUTS. Recall that C is the set of critical edges, R is the set of residual edges, and that $G_R = (V(G), R)$ is the residual graph. We construct the instance $(\mathbb{G}^*, d, k)$ of DIVERSE MIN s - t -CUTS of Lemma 6 as follows. We compute an s - t max-flow f . We contract all the edges in R that have positive flow through it.

After this, we contract all the strongly connected components of G_R' , where G_R' is the graph obtained from G_R by the above contractions. If multiple edges are formed, we replace them with one edge whose capacity (resp. multiplicity) is the sum of the capacities (resp. multiplicities) of these multiple edges. For the vertices obtained from s and t by the contraction, we keep the same names s and t , respectively. Let f^* be the flow obtained from f induced on G^* . As the residual edges with a positive flow were contracted, we can make the following remark.

► **Remark 12.** Edges in R have zero flow in the flow f^* .

The fact that $(\mathbb{G}, d, k)$ and $(\mathbb{G}^*, d, k)$ are equivalent instances of DIVERSE MIN s - t -CUTS follows from the next two lemmata. Informally, we first show that the minimum s - t cuts of G^* are exactly the minimum s - t cuts of G . The proof of the lemma is available in the full version of the paper [9].

► **Lemma 13.** *For every minimum s - t cut (A, B) of G , no edge with its endpoints in A and B is contracted in the construction of G^* , and (A^*, B^*) is a minimum s - t cut in G^* , where A^* and B^* are the sets obtained from A and B , respectively, by the contraction. Furthermore, for any minimum s - t cut (A^*, B^*) of G^* , (A, B) is a minimum s - t cut in G , where A^* and B^* are obtained from A and B .*

Lemma 13 establishes the bijection φ between minimum s - t cuts in G and G^* , where $\varphi(A, B) = (A^*, B^*)$. By the definition of the multiplicity function ℓ^* , we immediately obtain the following property.

► **Lemma 14.** *Let (A_1, B_1) and (A_2, B_2) be two minimum s - t cuts of G , and let $\varphi(A_1, B_1) = (A_1^*, B_1^*)$ and $\varphi(A_2, B_2) = (A_2^*, B_2^*)$. Then*

$$\ell(C_1 \triangle C_2) = \ell(C_1^* \triangle C_2^*),$$

where $C_i = E_G(A_i, B_i)$ and $C_i^* = E_{G^*}(A_i^*, B_i^*)$ for $i \in \{1, 2\}$.

This proves the equivalence of $(\mathbb{G}, d, k)$ and $(\mathbb{G}^*, d, k)$.

Since (G, d, k) and (G^*, d, k) are equivalent, from now on, we can assume that s is a source and t is a sink in G^* . Otherwise, we can simply delete the edges of the form (v, s) and (t, v) as they are irrelevant. Let $\tilde{G}$ be the graph obtained from G^* by reversing the direction of residual edges. Hence, the edge set of $\tilde{G}$ is $C \cup \tilde{R}$, where $\tilde{R}$ represent the reversed residual edges. We use $\tilde{G}$ to prove some nice properties of G^* .

► **Lemma 15.** *$\tilde{G}$ is a directed acyclic graph (DAG).*

Proof. Toward a contradiction, consider the existence of a directed cycle D . Given that all strongly connected components of G_R have been contracted, it implies that $(V(\tilde{G}), \tilde{R})$ is a DAG. Therefore, D contains at least one edge from C . Since e is critical in G^* , there exists a minimum s - t cut (A, B) in G^* containing e . By Observation 10, e is an edge from A to B . Since D is a cycle, there exists an edge $f \in D$ from B to A . By Observation 10, $f \notin C$ and thus $f \in \tilde{R}$. Thus, the edge $\tilde{f}$ corresponding to f in G^* is a residual edge from A to B .

Therefore $\tilde{f}$ belongs to (A, B) , contradicting the fact that $\tilde{f}$ is a residual edge. ◀

A *topological ordering* of a directed graph G is an ordering of the vertices of G , $v_1, \dots, v_n$, such that for every edge $v_i v_j$, $i < j$. It is well-known that G admits a topological ordering if and only if G is a DAG [3]. Using Lemma 15, we can prove the following.

► **Lemma 16.** *Let (A, B) be a minimum s - t cut in G^* . There is an ordering $s = v_1, v_2, \dots, v_n = t$ of $V(G^*)$ such that the following hold:*

1. *For each $1 \leq i \leq n$, $(\{v_1, \dots, v_i\}, \{v_{i+1}, \dots, v_n\})$ is a minimum s - t cut.*
2. *There exists $1 \leq j \leq n$ such that $A = \{v_1, \dots, v_j\}$ and $B = \{v_{j+1}, \dots, v_n\}$.*

Proof. Since (A, B) is a minimum s - t cut, all edges in $E(A, B)$ belong to C and all edges in $E(B, A)$ belong to R (from Observations 9 and 10). This means that in the graph $\tilde{G}$, $E(B, A) = \emptyset$. Since $\tilde{G}$ is a DAG by Lemma 15 and that no edge goes left through a minimum s - t cut, there exist a topological ordering of $\tilde{G}[A]$ and a topological ordering of $\tilde{G}[B]$ such that their concatenated ordering is a topological ordering $s = v_1, v_2, \dots, v_n = t$ of $\tilde{G}$. Notice that we can assume that s is the first and t is the last vertex in the ordering because s is a source and t is a sink. Hence, there exists $1 \leq j \leq n$ such that $A = \{v_1, \dots, v_j\}$ and $B = \{v_{j+1}, \dots, v_n\}$.

Let us now prove the first point. Let $1 \leq i \leq n$, $A_i = \{v_1, \dots, v_i\}$ and $B_i = \{v_{i+1}, \dots, v_n\}$. Clearly, $s \in A_i$ and $t \in B_i$, so (A_i, B_i) is an s - t -cut. By Lemma 4, $v(f) = \sum_{e \in E(A_i, B_i)} f(e) - \sum_{e \in E(B_i, A_i)} f(e)$. From the definition of the topological ordering, any edge in $e \in E(A_i, B_i)$ belongs to C and any edge in $e \in E(B_i, A_i)$ belongs to R and thus has zero flow by Remark 12. It follows that

$$\begin{aligned} v(f) &= \sum_{e \in E(A_i, B_i)} f(e) - \sum_{e \in E(B_i, A_i)} f(e) = \sum_{e \in E(A_i, B_i)} f(e) - 0 \\ &= \sum_{e \in E(A_i, B_i)} c(e) = c(A_i, B_i), \end{aligned}$$

so $\text{cut}(A_i, B_i)$ is an s - t -cut whose capacity is equal to the maximum flow, and thus it is a minimum s - t cut. ◀

3.3 Correctness of the Preprocessing

Using Lemma 16, we can prove that if G^* admits a minimum cut with many edges, then G^* is a yes-instance.

► **Lemma 17.** *Let $(\mathbb{G}, d, k)$ be an instance of DIVERSE MIN s - t -CUTS and $(\mathbb{G}^*, d, k)$ be the equivalent instance constructed in Section 3.2. If G^* has a minimum s - t cut with at least $(kd)^2$ edges, then $(\mathbb{G}^*, d, k)$ is a yes-instance of DIVERSE MIN s - t -CUTS. Moreover, we can output k d -diverse minimum s - t cuts in polynomial time.*

Proof. Let (A, B) be a minimum s - t cut of G^* with at least $(kd)^2$ edges and let $S = E_{G^*}(A, B)$. By Lemma 16, there exist an ordering $s = v_1, \dots, v_n = t$ of $V(G^*)$ and an index $1 \leq i \leq n$ such that $A = \{v_1, \dots, v_i\}$ and $B = \{v_{i+1}, \dots, v_n\}$.

Since G^* has no multiple edges, either the left side $\{v_1, \dots, v_i\}$ contains at least $d(k-1)+1$ endpoints of edges in S , or the right side $\{v_{i+1}, \dots, v_n\}$ does (indeed, if each side has at most $d(k-1) < kd$ endpoints, then we have $|S| < (dk)^2$). Assume w.l.o.g. the left side contains at least $d(k-1)+1$ endpoints. Let L be any set of $d(k-1)+1$ endpoints of S on the left side.

Define indices $i_1, \dots, i_k$ so that $i_1 < \dots < i_k$, i_1 is the minimum index of the vertices in L , and for each $j \in [2, k]$, v_{i_j} is the endpoint in L such that exactly $d-1$ vertices of L are between $v_{i_{j-1}}$ and v_{i_j} in the ordering. Since $|L| \geq d(k-1)+1$, it implies that $i_k \leq i$. For each $j \in [k]$, we set $A_j = \{v_1, \dots, v_{i_j}\}$ and $B_j = \{v_{i_j+1}, \dots, v_n\}$ (see Figure 1). By Lemma 16, each (A_j, B_j) is a minimum s - t cut. For every $j, j' \in [k]$ such that $j < j'$, we have that $E_{G^*}(A_{j'}, B_{j'})$ contains at least d edges of S that are not in $E_{G^*}(A_j, B_j)$. Thus, for $C_j = E_{G^*}(A_j, B_j)$ and $C_{j'} = E_{G^*}(A_{j'}, B_{j'})$, $|C_{j'} \Delta C_j| \geq d$. This means that $C_1, \dots, C_k$ are d -diverse.

Therefore, if $|S| \geq (dk)^2$, the instance is a yes-instance. ◀

We are finally ready to prove the main preprocessing lemma.

Proof of Lemma 6. We find a maximum s - t flow in G in polynomial time. Then by making use Observation 9, we check for every edge e , whether it is critical by calling the flow algorithm for $G - e$. This way, we construct the sets C and R in $\mathcal{O}(m^{2+o(1)})$ time. In the next step, we contract the residual edges that have positive flow through them. We then compute in linear time the strongly connected components of G'_R [24], and contract them. This way, in time $\mathcal{O}(m^{2+o(1)})$, we construct the instance $(\mathbb{G}^*, d, k)$ as described in Section 3.2. By Lemma 13, $(\mathbb{G}, d, k)$ and $(\mathbb{G}^*, d, k)$ are equivalent instances of DIVERSE MIN s - t -CUTS. In time $\mathcal{O}(m^{1+o(1)})$, we apply Lemma 7 with input $\mathbb{G}^*$. It outputs a minimum s - t cut (A, B) of G^* whose number of edges is maximized. If (A, B) has at least $(kd)^2$ edges, then, by Lemma 17, $(\mathbb{G}^*, d, k)$, and thus $(\mathbb{G}, d, k)$, is a yes-instance of DIVERSE MIN s - t -CUTS. Otherwise, we return $(\mathbb{G}^*, d, k)$, given that any minimum s - t cut of G^* has at most $(kd)^2$ edges. ◀

4 Algorithm for Solving Bounded Cuts

In this section, we restate and prove Lemma 18.

► **Lemma 18.** *There is an algorithm that solves DIVERSE MIN s - t -CUTS in time $2^{k^2 r} \cdot (dkr)^{\mathcal{O}(kr)} \cdot r^{\mathcal{O}(r^4)} \cdot mn \log m$, where r is the maximum number of edges of a minimum s - t cut of the input graph.*

4.1 Color Coding

We now describe the color-coding scheme that will allow us to isolate the relevant cut structure. Let us begin with some definitions. Let $(\mathbb{G}, d, k)$ be a yes-instance of DIVERSE MIN s - t -CUTS and let $\mathcal{C} = (C_1, \dots, C_k)$ be d -diverse minimum s - t cuts in G . Let $U = C_1 \cup \dots \cup C_k$.

Membership encoding. Each edge $e \in E$ has a membership vector

$$\beta(e) \in \{0, 1\}^k, \quad \beta(e)_i = 1 \iff e \in C_i.$$

The set of edges of U corresponding to $\alpha \in \{0, 1\}^k$ is defined by

$$C_\alpha^\cap := \{e \in U : \beta(e) = \alpha\}.$$

Active regions. We define the set of *active regions* of $\mathcal{C}$ by

$$\mathcal{R}_{\text{act}} := \{\alpha \in \{0, 1\}^k : C_\alpha^\cap \neq \emptyset\}.$$

That is, $\alpha \in \mathcal{R}_{\text{act}}$ if and only if region $C_\alpha^\cap$ contains at least one edge that belongs to at least one of the cuts in $\mathcal{C}$.

► **Observation 19.** Assume every minimum s - t cut C in G satisfies $|C| \leq r$. Then the family $\mathcal{C} = (C_1, \dots, C_k)$ has at most kr active regions:

$$|\mathcal{R}_{\text{act}}| \leq kr.$$

Note that the number of choices for the set of active regions is thus at most $(2^k)^{kr} = 2^{k^2 r}$.

Colorings. We use color coding with kr colors, identified with vectors $\alpha \in \mathcal{R}_{\text{act}}$. A coloring of G is a function $\chi : E \rightarrow \mathcal{R}_{\text{act}}$. For each $\alpha \in \mathcal{R}_{\text{act}}$, the coloring induces the region (color class) $E_\alpha := \{e \in E : \chi(e) = \alpha\}$. A coloring χ is *good with respect to $\mathcal{C}$* if it agrees with the membership vectors of all edges in U , i.e., for each $e \in U$, we have that $\chi(e) = \beta(e)$, or, said another way, for every $\alpha \in \{0, 1\}^k$, we have $C_\alpha^\cap = E_\alpha \cap U$.

Number of edges and multiplicity of each region. For each active region α , we set

$$x_\alpha = \sum_{e \in C_\alpha^\cap} \ell(e) \quad \text{and} \quad \text{if } x_\alpha \geq d, \text{ we set } x_\alpha = d^+ \text{ instead}$$

So we have $x_\alpha \in \{0, 1, \dots, d-1, d^+\} =: [0, d^+]$. Similarly, we set r_α to be the number of edges in a region $C_\alpha^\cap$. Since the number of edges in each cut is bounded by r , $r_\alpha \in [0, r]$. We call *profile of $\mathcal{C}$* the collection $\mathcal{P} := \{(x_\alpha, r_\alpha) : \alpha \in \mathcal{R}_{\text{act}}\}$.

More generally, we call *profile with respect to $\mathcal{R}_{\text{act}}$* any collection $\mathcal{P} := \{(l_\alpha, n_\alpha) : \alpha \in \mathcal{R}_{\text{act}}, l_\alpha \in [0, d^+], n_\alpha \in [0, r]\}$. Note that the number of profiles with respect to $\mathcal{R}_{\text{act}}$ is thus at most $((r+1)(d+1))^{kr}$.

For convenience, we set $x_\alpha = r_\alpha = 0$ and $E_\alpha = \emptyset$ for each $\alpha \in \{0, 1\}^k \setminus \mathcal{R}_{\text{act}}$, and we call *profile* any profile with respect to some set of active regions $\mathcal{R}_{\text{act}}$.

Note that by the d -diversity of $\mathcal{C}$, the multiplicity summation across regions where exactly either C_1 or C_2 intersects (for any two cuts $C_i, C_j \in \mathcal{C}$) takes value at least d . Formally, we have the following:

► **Observation 20.** For each $i, j \in [k]$ such that $i \neq j$, $\sum_{\alpha: \alpha_i \neq \alpha_j} x_\alpha \geq d$.

The main result from this section and this part of the algorithm is the following lemma on deterministic guessing. We apply a deterministic version of color coding as follows to guess a coloring that is good with respect to $\mathcal{C}$ and its profile.

► **Lemma 21.** *Let (G, d, k) be an instance of DIVERSE MIN s - t CUTS, let $\mathcal{R}_{\text{act}}$ be a set of active regions and let $\mathcal{P}$ be a profile with respect to $\mathcal{R}_{\text{act}}$. Then one can construct in time $(kr)^{\mathcal{O}(kr)} m \log m$ a set $\mathcal{H}$ of colorings of G of size at most $(kr)^{\mathcal{O}(kr)} \log m$ such that, for every d -diverse minimum s - t cuts $\mathcal{C} = (C_1, \dots, C_k)$ in G with profile $\mathcal{P}$, there exists a coloring $\chi \in \mathcal{H}$ that is good with respect to $\mathcal{C}$.*

Proof. We apply a deterministic version of color coding using standard splitter constructions; we refer to the textbook on parameterized algorithms [6] for the introduction. In particular, by known results on (n, s) -perfect hash families, one can find in time $e^s \cdot s^{\mathcal{O}(\log s)} m \log m$ an explicit family $\mathcal{F}$ of functions $\lambda : E \rightarrow [s]$ of size $e^s \cdot s^{\mathcal{O}(\log s)} m \log m$ such that for every subset $D \subseteq E$ of size at most s , there exists some $\lambda \in \mathcal{F}$ such that λ is bijective on D . In particular, for $s = kr$, this is the case for $D = \bigcup_{i \in [k]} C_i$ for any minimum s - t cuts $(C_1, \dots, C_k)$. Let $\mathcal{G}$ represent the set of all functions from $[s]$ to $\mathcal{R}_{\text{act}}$ and let $\mathcal{H} := \mathcal{G} \circ \mathcal{F} = \{g \circ f \mid g \in \mathcal{G}, f \in \mathcal{F}\}$. We claim that the statement holds with the family $\mathcal{H}$. Let $\mathcal{C} = (C_1, \dots, C_k)$ be d -diverse minimum s - t cuts in G with profile $\mathcal{P}$. Since at most kr edges belong to $U := \bigcup_{i \in [k]} C_i$, there is a function $f \in \mathcal{F}$ that is bijective from U to $[kr]$ and furthermore, there exists a function $g \in \mathcal{G}$ that labels these edges to the correct active region in $\mathcal{R}_{\text{act}}$. Hence $\chi = g \circ f$ is a coloring $\chi \in \mathcal{H}$ that is good with respect to $\mathcal{C}$.

The size of $\mathcal{H}$ is bounded by $(kr)^{kr} \cdot e^{kr} \cdot (kr)^{\mathcal{O}(\log kr)} \log m$ and takes time $(kr)^{kr} \cdot e^{kr} \cdot (kr)^{\mathcal{O}(\log kr)} \log m$ to compute. Upon simplification, we obtain $|\mathcal{H}| = (kr)^{\mathcal{O}(kr)} \log m$, and $\mathcal{H}$ is computed in time $(kr)^{\mathcal{O}(kr)} m \log m$. ◀

4.2 Flow Augmentation

The *flow augmentation* technique designed by Kim et al. [18] transforms minimal cuts into minimum cuts. The blackbox result (in its deterministic version) is as follows:

► **Theorem 22** (Flow Augmentation [18]). *There exists an algorithm that, given a directed graph G , two vertices $s, t \in V(G)$, and an integer λ , in time $2^{\mathcal{O}(\lambda^4 \log \lambda)} n^{\mathcal{O}(1)}$ outputs a set $\mathcal{A} \subseteq 2^{V(G) \times V(G)}$ of size $2^{\mathcal{O}(\lambda^4 \log \lambda)} (\log n)^{\mathcal{O}(\lambda^3)}$ such that for every minimal s - t cut $Z \subseteq E(G)$ of size at most λ there exists $A \in \mathcal{A}$ such that Z remains an s - t cut in $G + A$ and, furthermore, Z is a minimum s - t cut in $G + A$.*

Note that, given that $\lambda^3 \log \log n \leq \lambda^4$ if $\log \log n \leq \lambda$ and $\lambda^3 \log \log n \leq (\log \log n)^4$ otherwise, we have

$$\begin{aligned} |\mathcal{A}| &= 2^{\mathcal{O}(\lambda^4 \log \lambda)} (\log n)^{\mathcal{O}(\lambda^3)} = 2^{\mathcal{O}(\lambda^4 \log \lambda)} \cdot 2^{\mathcal{O}(\lambda^3 \log \log n)} \\ &\leq 2^{\mathcal{O}(\lambda^4 \log \lambda)} \cdot 2^{\mathcal{O}(\lambda^4 + (\log \log n)^4)} = 2^{\mathcal{O}(\lambda^4 \log \lambda)} n^{o(1)}. \end{aligned}$$

We use flow augmentation to find minimum cuts with the required colors and specifications.

► **Lemma 23.** *There is an algorithm that, given a directed graph G , an edge capacity c , a coloring χ of G , a profile $\mathcal{P} := \{(x_\alpha, r_\alpha) : \alpha \in \{0, 1\}^k\}$, and an integer r , find in time $k^{r+1} r^{\mathcal{O}(r^4)} d^r mn$, if they exist, for each $i \in [k]$, a minimum cut C_i' such that,*

- $C_i' \subseteq \bigcup_{\alpha: \alpha_i = 1} E_\alpha$ and $\lambda_i = |C_i'| \leq r$, and
- for each active α , $\sum_{e \in C_i' \cap E_\alpha} \ell(e) = x_\alpha$ and $|C_i' \cap E_\alpha| = r_\alpha$.

Proof. Let $\mathbf{1} : E(G) \rightarrow \{1\}$ be the all-ones capacity function. The deterministic version of flow augmentation (Theorem 22) guarantees a set $\mathcal{A}$ such that for every C that is minimal in $(G, \mathbf{1})$, there exists $A \in \mathcal{A}$ such that C is an minimum s - t cut in $(G + A, \mathbf{1})$. For $(G, \mathbf{1})$, use Theorem 22 to obtain $\mathcal{A}$.

We take a step back to see why minimal cuts come into the picture.

▷ **Claim 24.** If C is a minimum s - t cut in (G, c) , then C is a minimal s - t cut in $(G, \mathbf{1})$.

Proof. A minimum s - t cut is inclusion-wise minimal. If C is not minimal, let $C' \subset C$ such that C' is an s - t minimal cut in $(G, \mathbf{1})$. This means that C' is a cut in (G, c) such that $c(C') < c(C)$. This contradicts the fact that C is an s - t mincut. ◁

Onto the procedure, for each $i \in [k]$, the iteration consists of the following steps.

Step 1 (The augmentation of edges for the cut). Since C'_i is minimal in $(G, \mathbf{1})$ by Claim 24, we can find $A_i \in \mathcal{A}$ such that C'_i is a minimum s - t cut in $G + A_i$. This A_i can be found in $2^{\mathcal{O}(\lambda_i^4 \log \lambda_i)} n^{o(1)}$ guesses by Theorem 22. From our bounds on λ_i , this is $2^{\mathcal{O}(r^4 \log r)} n^{o(1)}$.

Step 2 (Menger paths). Recall that λ_i represents the number of edges expected in C'_i , which is a minimum s - t cut in $(G + A_i, \mathbf{1})$. By Lemma 5, we construct $P_1, \dots, P_{\lambda_i}$, the edge-disjoint s - t paths in $G + A_i$.

Step 3 (Assigning types to paths). A minimum s - t cut takes one edge from each s - t path P_j . Each edge in C'_i is painted with a color and multiplicity $(\chi(e), \ell(e))$. This means that we are required to guess the color and the multiplicity of the edge from each path P_j . The multiplicity can take values from $\{1, \dots, d-1, d^+\}$ and there are at most kr colors to choose from.

Naturally, guesses that do not satisfy the conditions of Lemma 23 are discarded. The relevant conditions to be met on the guesses are highlighted below. For $1 \leq j \leq \lambda_i$,

1. $(\alpha_j)_i = 1$ (so that $C'_i \subseteq \bigcup_{\alpha: \alpha_i=1} E_\alpha$),
2. for all active α such that $(\alpha)_i = 1$, if $x_\alpha = d^+$ then $\sum_{j: \alpha_j=\alpha} l_j \geq d$ and otherwise $\sum_{j: \alpha_j=\alpha} l_j = x_\alpha$ (so that $\sum_{e \in C'_i \cap E_\alpha} \ell(e) = x_\alpha$), and
3. For all active α such that $(\alpha)_i = 1$, $|\{j : \alpha_j = \alpha\}| = r_\alpha$ (so that $|C'_i \cap E_\alpha| = r_\alpha$).

Step 4 (Capacity constraints to compute a cut). We assign capacities to the graph $G + A_i$. Define $c_i(e) = c(e)$ if $e \in P_j$ for some j and if $(\chi(e), \ell(e)) = (\alpha_j, l_j)$. Otherwise, set $c_i(e) = +\infty$. The graph $(G + A_i, c_i)$ is the auxiliary graph.

Step 5 (Reconstruction of the cuts). Compute a minimum s - t cut C'_i in $(G + A_i, c_i)$ and check if its capacity equals μ . If not, return *no*-instance. Given that C'_i is a minimum s - t cut in $(G + A_i, c_i)$, it is also a minimum s - t cut in (G, c) by construction. Additionally, since bad guesses were discarded at Step 3, we know that C'_i follows the requirements of the statement.

Hence, after going through Steps 1–5 for each $i \in [k]$, if $C'_1, \dots, C'_k$ are the resulting cuts then we report a *yes*-instance and output these cuts. Otherwise, we report a *no*-instance.

Correctness. Suppose that there exist $C_1, \dots, C_k$ following the requirements of the statement. Let us argue that the algorithm returns a *yes*-instance. Consider the graph $(G, \mathbf{1})$. By Claim 24, each C_i is a minimal cut in $(G, \mathbf{1})$. In Step 1, by Theorem 22, there exists a set $\mathcal{A}$ such that for some $A_i \in \mathcal{A}$, the cut C_i is a minimum cut in $G + A_i$. The procedure enumerates all such A_i , hence it considers the correct one. Step 2 constructs $P_1, \dots, P_{\lambda_i}$ edge disjoint s - t paths. We know that C_i takes exactly one edge from each path P_j . Let

edge e_j denote the edge selected from P_j by C_i . In Step 3, the individual color and the multiplicity $(\chi(e_j), \ell(e_j))$ is guessed for each path P_j and this guess is relevant and is not discarded since conditions 1, 2 and 3 are satisfied. In Step 4, for a path P_j , only the edges with color and multiplicity $(\chi(e_j), \ell(e_j))$ have their finite capacities c assigned. The rest of the edges have infinite capacities. Under this assignment, C_i remains feasible. In Step 5, we know that C_i satisfies as a minimum cut for $(G + A_i, c_i)$. Indeed, a cut with lower capacity would also be a cut in (G, c) , a contradiction. Let C'_i be the cut obtained by running the minimum cut algorithm on $(G + A_i, c_i)$. Repeating these steps for each $i \in [k]$, we obtain $C'_1, \dots, C'_k$ and this is output by the algorithm.

Running time. The flow augmentation requires to be run only once and computation of $\mathcal{A}$ takes time $2^{(r)^4 \log r} n^{\mathcal{O}(1)}$. For each $i \in [k]$, the running time is as follows. We know that $|\mathcal{A}| \leq 2^{(r)^4 \log r} n^{\mathcal{O}(1)}$. We guess $A_i \in \mathcal{A}$, and then compute Menger paths in time $\mathcal{O}(m\sqrt{n})$ (see Lemma 5). The assignment of types to the paths takes at most $(kr)d$ guesses for each path and there can be at most r paths in total. This takes $\mathcal{O}((krd)^r)$ time in total. Following this, the minimum s - t cut computation for each guess can be done in time $\mathcal{O}(m^{1+o(1)})$ [5]. In total, we run this procedure over k iterations and end up using time

$$2^{\mathcal{O}(r^4 \log(r))} \cdot k \cdot n + 2^{\mathcal{O}(r^4 \log(r))} \cdot k \cdot (krd)^r \cdot \mathcal{O}(m\sqrt{n}).$$

This simplifies to $\mathcal{O}(k \cdot 2^{r^4 \log(r)} \cdot (krd)^r \cdot m\sqrt{n})$. Upon simplifying logarithms and exponentials, the running time is $\mathcal{O}(k^{r+1} r^{\mathcal{O}(r^4)} d^r m\sqrt{n})$. ◀

► **Lemma 25.** *Let χ be a coloring of G and $\mathcal{P} := \{(x_\alpha, r_\alpha) : \alpha \in \{0, 1\}^k\}$ be a profile such that, for each $i, j \in [k]$, $\sum_{\alpha: \alpha_i \neq \alpha_j} x_\alpha \geq d$. Then the cuts $C'_1, \dots, C'_k$ output by Lemma 23 are d -diverse.*

Proof. Let us show that C'_i and C'_j are d -diverse for each $i \neq j$. We have:

$$\begin{aligned} \ell(C'_i \triangle C'_j) &\geq \sum_{\alpha: \alpha_i=1, \alpha_j=0} \sum_{e \in C'_i \cap E_\alpha} \ell(e) + \sum_{\alpha: \alpha_i=0, \alpha_j=1} \sum_{e \in C'_j \cap E_\alpha} \ell(e) \\ &= \sum_{\alpha: \alpha_i=1, \alpha_j=0} x_\alpha + \sum_{\alpha: \alpha_i=0, \alpha_j=1} x_\alpha = \sum_{\alpha: \alpha_i \neq \alpha_j} x_\alpha \geq d \end{aligned} \quad \blacktriangleleft$$

4.3 Correctness of the Algorithm

We can now prove Lemma 18.

Proof of Lemma 18. Our algorithm goes as follows. For each set of active regions $\mathcal{R}_{\text{act}}$ (at most $2^{k^2 r}$ such sets) and for each profile $\mathcal{P}$ with respect to $\mathcal{R}_{\text{act}}$ (at most $((r+1)(d+1))^{kr}$ such profiles), we do the following. Firstly, we only retain the profiles for which Observation 20 holds and discard the rest. Using Lemma 21, we construct in time $(kr)^{\mathcal{O}(kr)} m \log m$ a set $\mathcal{H}$ of colorings of G of size at most $(kr)^{\mathcal{O}(kr)} \log m$ such that, for every d -diverse minimum s - t cuts $\mathcal{C} = (C_1, \dots, C_k)$ in G with profile $\mathcal{P}$, there exists a coloring $\chi \in \mathcal{H}$ that is good with respect to $\mathcal{C}$.

For each coloring $\chi \in \mathcal{H}$, we invoke the procedure of Lemma 23 to test whether there exists a family of cuts $C'_1, \dots, C'_k$ such that, for each $i \in [k]$,

- $C'_i \subseteq \bigcup_{\alpha: \alpha_i=1} E_\alpha$ and $\lambda_i := |C'_i| \leq r$, and
- $\sum_{e \in C'_i \cap E_\alpha} \ell(e) = x_\alpha$ and $|C'_i \cap E_\alpha| = r_\alpha$.

If any call succeeds, output the corresponding cuts. If no profile yields a solution, report that no such family of cuts exists.

Correctness. If (G, d, k) is a yes-instance, then there exists k d -diverse minimum s - t cuts $(C_1, \dots, C_k)$. At some point in the algorithm, we consider the set of active regions $\mathcal{R}_{\text{act}}$ of $(C_1, \dots, C_k)$ and the profile $\mathcal{P} = \{(x_\alpha, r_\alpha) : \alpha \in \mathcal{R}_{\text{act}}\}$ of $(C_1, \dots, C_k)$. Additionally, by Lemma 21, there is a coloring χ of G in $\mathcal{H}$ that is good with respect to $(C_1, \dots, C_k)$. Hence, by Lemma 23, we should find k cuts $C'_1, \dots, C'_k$ with the correct requirement (since $C_1, \dots, C_k$ follows those requirements). Since Observation 20 holds, for each $i \in [k]$, we have $\sum_{\alpha: \alpha_i \neq \alpha_j} x_\alpha \geq d$. This implies, by Lemma 25, that $C'_1, \dots, C'_k$ are d -diverse. Hence the result.

Running time. From Lemma 21 and Lemma 23, the total running time is:

$$2^{\mathcal{O}(k^2 r)} \cdot ((r+1)(d+1))^{kr} \cdot \left((kr)^{\mathcal{O}(kr)} m \log m + (kr)^{\mathcal{O}(kr)} \log m \cdot k^{r+1} r^{\mathcal{O}(r^4)} d^r m \sqrt{n} \right)$$

which, after simplification, gives $\mathcal{O}(2^{\mathcal{O}(k^2 r)} \cdot (dkr)^{\mathcal{O}(kr)} \cdot r^{\mathcal{O}(r^4)} \cdot m \sqrt{n} \log n)$. $\blacktriangleleft$

5 Conclusion

We conclude by stating some open problems. In our paper, we proved that the DIVERSE MIN s - t -CUTS problem is FPT when parameterized by k and d . However, the running time of our algorithm is superexponential. It is natural to ask whether it is possible to do better. The existing complexity lower bound of de Berg, López Martínez, and Spieksma [7] does not exclude existence of single-exponential in k and d algorithm. Furthermore, it is interesting to know whether we can get a substantially better running time for the problem on undirected graphs, as this case may be easier. In particular, we can contract all residual edges in the preprocessing step. Then for the obtained instance, it can be shown that either it is a yes-instance of the problem or the pathwidth of the graph does not exceed kd . Another natural question is whether DIVERSE MIN s - t -CUTS admits a polynomial kernel for the parameterization by k and d .

Further questions concern lower bounds. For $d \in \{1, 2\}$, DIVERSE MIN s - t -CUTS is equivalent to deciding whether the input weighted graph has k distinct minimum s - t cuts. Since the problem of counting minimum s - t cuts is #P-complete by the result of Provan and Ball [22], we obtain that DIVERSE MIN s - t -CUTS is intractable even if $d \in \{1, 2\}$. However, this is not the case when k is polynomial in the graph size because minimum s - t cuts can be enumerated with polynomial (linear) delay [23, 25]. Is DIVERSE MIN s - t -CUTS NP-complete in the strong sense for a given constant $d \geq 3$, that is, when k is polynomial in n ?

Finally, what can be said for the dual parameterization, that is, when the diversity measure is the maximum size of the intersection of two cuts? Is the problem FPT in this case? Notice that the maximum family of disjoint s - t -cuts can be found in polynomial time by the results of Wagner [26].

References

- 1 Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., USA, 1993.
- 2 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.
- 3 Jørgen Bang-Jensen and Gregory Z. Gutin. *Digraphs - Theory, Algorithms and Applications, Second Edition*. Springer Monographs in Mathematics. Springer, 2009.

- 4 Julien Baste, Michael R. Fellows, Lars Jaffke, Tomáš Masařík, Mateus de Oliveira Oliveira, Geevarghese Philip, and Frances A. Rosamond. Diversity of solutions: An exploration through the lens of fixed-parameter tractability theory. *Artificial Intelligence*, 303:103644, 2022. doi:10.1016/j.artint.2021.103644.
- 5 Li Chen, Rasmus Kyng, Yang Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. *Journal of the ACM*, 72(3):1–103, 2025. doi:10.1145/3728631.
- 6 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*, volume 4. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 7 Mark de Berg, Andrés López Martínez, and Frits Spieksma. Finding Diverse Minimum s - t Cuts. *LIPIcs, Volume 283, ISAAC 2023*, 283:24:1–24:17, 2023. doi:10.4230/LIPIcs.ISAAC.2023.24.
- 8 Mark de Berg, Andrés López Martínez, and Frits Spieksma. Disjoint tours and the price of diversity. In *WALCOM: Algorithms and Computation*, pages 402–417, Singapore, 2026. Springer Nature. doi:10.1007/978-981-95-7127-7_27.
- 9 Krishnan Dehaleesan, Pål Grønås Drange, Fedor V. Fomin, Petr A. Golovach, and Laure Morelle. An FPT algorithm for diverse minimum s - t cuts, 2026. arXiv:2607.03266.
- 10 Reinhard Diestel. *Graph Theory 5th edition. Graduate texts in mathematics*, volume 173. Springer, 2016. doi:10.1007/978-3-662-53622-3.
- 11 Karolina Drabik and Tomáš Masařík. Finding diverse solutions parameterized by cliquewidth. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 36864–36872, 2026. doi:10.1609/aaai.v40i43.41013.
- 12 Fedor V. Fomin, Petr A. Golovach, Lars Jaffke, Geevarghese Philip, and Danil Sagunov. Diverse pairs of matchings. *Algorithmica*, 86(6):2026–2040, 2024. doi:10.1007/s00453-024-01214-7.
- 13 Fedor V. Fomin, Petr A. Golovach, Fahad Panolan, Geevarghese Philip, and Saket Saurabh. Diverse collections in matroids and graphs. *Mathematical Programming*, 204(1):415–447, 2024. doi:10.1007/s10107-023-01959-z.
- 14 Tesshu Hanaka, Masashi Kiyomi, Yasuaki Kobayashi, Yusuke Kobayashi, Kazuhiro Kurita, and Yota Otachi. A framework to design approximation algorithms for finding diverse solutions in combinatorial problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4):3968–3976, 2023. doi:10.1609/aaai.v37i4.25511.
- 15 Tesshu Hanaka, Yasuaki Kobayashi, Kazuhiro Kurita, See Woo Lee, and Yota Otachi. Computing diverse shortest paths efficiently: A theoretical and experimental study. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4):3758–3766, 2022. doi:10.1609/aaai.v36i4.20290.
- 16 Tesshu Hanaka, Yasuaki Kobayashi, Kazuhiro Kurita, and Yota Otachi. Finding diverse trees, paths, and more. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):3778–3786, 2021. doi:10.1609/aaai.v35i5.16495.
- 17 David R. Karger and Clifford Stein. A new approach to the minimum cut problem. *Journal of the ACM*, 43(4):601–640, 1996. doi:10.1145/234533.234534.
- 18 Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlström. Flow-augmentation I: Directed graphs. *Journal of the ACM*, 72(1):1–38, 2025. doi:10.1145/3706103.
- 19 Jon Kleinberg and Éva Tardos. *Algorithm design*. Pearson, 2006.
- 20 Ching-Chung Kuo, Fred Glover, and Krishna S. Dhir. Analyzing and modeling the maximum diversity problem by zero-one programming. *Decision Sciences*, 24(6):1171–1185, 1993. doi:10.1111/j.1540-5915.1993.tb00509.x.
- 21 Jean-Claude Picard and Maurice Queyranne. On the structure of all minimum cuts in a network and applications. In *Combinatorial Optimization II*, pages 8–16. Springer Berlin Heidelberg, Berlin, Heidelberg, 1980. doi:10.1007/BFb0120902.

- 22 J. Scott Provan and Michael O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4):777–788, 1983. doi:10.1137/0212053.
- 23 J. Scott Provan and Douglas R. Shier. A paradigm for listing (s, t) -cuts in graphs. *Algorithmica*, 15(4):351–372, 1996. doi:10.1007/BF01961544.
- 24 Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972. doi:10.1137/0201010.
- 25 S. Tsukiyama, I. Shirakawa, H. Ozaki, and H. Ariyoshi. An algorithm to enumerate all cutsets of a graph in linear time per cutset. *J. ACM*, 27(4):619–632, 1980. doi:10.1145/322217.322220.
- 26 Donald K. Wagner. Disjoint (s, t) -cuts in a network. *Networks*, 20(4):361–371, 1990. doi:10.1002/net.3230200402.