

FPT Inapproximability of Directed Cut and Connectivity Problems

Rajesh Chitnis

School of Computer Science, University of Birmingham, UK
rajeshchitnis@gmail.com

Andreas Emil Feldmann

Charles University, Czechia
feldmann.a.e@gmail.com

Abstract

Cut problems and connectivity problems on digraphs are two well-studied classes of problems from the viewpoint of parameterized complexity. After a series of papers over the last decade, we now have (almost) tight bounds for the running time of several standard variants of these problems parameterized by two parameters: the number k of terminals and the size p of the solution. When there is evidence of FPT intractability, then the next natural alternative is to consider FPT approximations. In this paper, we show two types of results for directed cut and connectivity problems, building on existing results from the literature: first is to circumvent the hardness results for these problems by designing FPT approximation algorithms, or alternatively strengthen the existing hardness results by creating “gap-instances” under stronger hypotheses such as the (Gap-)Exponential Time Hypothesis (ETH). Formally, we show the following results:

Cutting paths between a set of terminal pairs, i.e., Directed Multicut: Pilipczuk and Wahlström [TOCT ’18] showed that DIRECTED MULTICUT is W[1]-hard when parameterized by p if $k = 4$. We complement this by showing the following two results:

- DIRECTED MULTICUT has a $k/2$ -approximation in $2^{O(p^2)} \cdot n^{O(1)}$ time (i.e., a 2-approximation if $k = 4$),
- Under Gap-ETH, DIRECTED MULTICUT does not admit an $(\frac{59}{58} - \epsilon)$ -approximation in $f(p) \cdot n^{O(1)}$ time, for any computable function f , even if $k = 4$.

Connecting a set of terminal pairs, i.e., Directed Steiner Network (DSN): The DSN problem on general graphs is known to be W[1]-hard parameterized by $p + k$ due to Guo et al. [SIDMA ’11]. Dinur and Manurangsi [ITCS ’18] further showed that there is no FPT $k^{1/4 - o(1)}$ -approximation algorithm parameterized by k , under Gap-ETH. Chitnis et al. [SODA ’14] considered the restriction to special graph classes, but unfortunately this does not lead to FPT algorithms either: DSN on planar graphs is W[1]-hard parameterized by k . In this paper we consider the $\text{DSN}_{\text{PLANAR}}$ problem which is an intermediate version: the graph is general, but we want to find a solution whose cost is at most that of an optimal planar solution (if one exists). We show the following lower bounds for $\text{DSN}_{\text{PLANAR}}$:

- $\text{DSN}_{\text{PLANAR}}$ has no $(2 - \epsilon)$ -approximation in FPT time parameterized by k , under Gap-ETH. This answers in the negative a question of Chitnis et al. [ESA ’18].
- $\text{DSN}_{\text{PLANAR}}$ is W[1]-hard parameterized by $k + p$. Moreover, under ETH, there is no $(1 + \epsilon)$ -approximation for $\text{DSN}_{\text{PLANAR}}$ in $f(k, p, \epsilon) \cdot n^{o(\sqrt{k+p+1/\epsilon})}$ time for any computable function f .

Pairwise connecting a set of terminals, i.e., Strongly Connected Steiner Subgraph (SCSS):

Guo et al. [SIDMA ’11] showed that SCSS is W[1]-hard parameterized by $p + k$, while Chitnis et al. [SODA ’14] showed that SCSS remains W[1]-hard parameterized by p , even if the input graph is planar. In this paper we consider the $\text{SCSS}_{\text{PLANAR}}$ problem which is an intermediate version: the graph is general, but we want to find a solution whose cost is at most that of an optimal planar solution (if one exists). We show the following lower bounds for $\text{SCSS}_{\text{PLANAR}}$:

- $\text{SCSS}_{\text{PLANAR}}$ is W[1]-hard parameterized by $k + p$. Moreover, under ETH, there is no $(1 + \epsilon)$ -approximation for $\text{SCSS}_{\text{PLANAR}}$ in $f(k, p, \epsilon) \cdot n^{o(\sqrt{k+p+1/\epsilon})}$ time for any computable function f .

Previously, the only known FPT approximation results for SCSS applied to general graphs parameterized by k : a 2-approximation by Chitnis et al. [IPEC ’13], and a matching $(2 - \epsilon)$ -hardness under Gap-ETH by Chitnis et al. [ESA ’18].



© Rajesh Chitnis and Andreas Emil Feldmann;
licensed under Creative Commons License CC-BY

th International Symposium on Parameterized and Exact Computation (IPEC 2019).

Editors: Bart M. P. Jansen and Jan Arne Telle; Article No. 8; pp. 8:1–8:20

Leibniz International Proceedings in Informatics



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

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Directed graphs, cuts, connectivity, Steiner problems, FPT inapproximability

Digital Object Identifier 10.4230/LIPIcs.IPEC.2019.8

Related Version Full version of the paper is available at <http://arxiv.org/abs/1910.01934>.

Funding *Rajesh Chitnis*: Work done while at University of Warwick, UK and supported by ERC grant 2014-CoG 647557.

Andreas Emil Feldmann: Supported by the Czech Science Foundation GAČR (grant #19-27871X), and by the Center for Foundations of Modern Computer Science (Charles Univ. project UNCE/S-CI/004).

Acknowledgements We thank Pasin Manurangsi for helpful discussions.

1 Introduction

Given a weighted directed graph $G = (V, E)$ with two terminal vertices s, t the problems of finding a minimum weight $s \rightsquigarrow t$ cut and a minimum weight $s \rightsquigarrow t$ path can both be famously solved in polynomial time. There are two natural generalizations when we consider more than two terminals: either we look for connectivity/cuts between all terminals of a given set, or we look for connectivity/cuts between a given set of terminal pairs. This leads to the four problems of DIRECTED MULTIWAY CUT, DIRECTED MULTICUT, STRONGLY CONNECTED STEINER SUBGRAPH and DIRECTED STEINER NETWORK:

- **Cutting all paths between a set of terminals:** In the DIRECTED MULTIWAY CUT problem, we are given a set of terminals $T = \{t_1, t_2, \dots, t_k\}$ and the goal is to find a minimum weight subset $X \subseteq V$ such that $G \setminus X$ has no $t_i \rightsquigarrow t_j$ path for any $1 \leq i \neq j \leq k$.
- **Cutting paths between a set of terminal pairs:** In the DIRECTED MULTICUT problem, we are given a set of terminal pairs $T = \{(s_i, t_i)\}_{i=1}^k$ and the goal is to find a minimum weight subset $X \subseteq V$ such that $G \setminus X$ has no $s_i \rightsquigarrow t_i$ path for any $1 \leq i \leq k$.
- **Connecting all terminals of a given set:** In the STRONGLY CONNECTED STEINER SUBGRAPH (SCSS) problem, we are given a set of terminals $T = \{t_1, t_2, \dots, t_k\}$ and the goal is to find a minimum weight subset $X \subseteq V$ such that $G[X]$ has a $t_i \rightsquigarrow t_j$ path for every $1 \leq i \neq j \leq k$.
- **Connecting a set of terminal pairs:** In the DIRECTED STEINER NETWORK (DSN) problem, we are given a set of terminal pairs $T = \{(s_i, t_i)\}_{i=1}^k$ and the goal is to find a minimum weight subset $X \subseteq V$ such that $G[X]$ has an $s_i \rightsquigarrow t_i$ path for every $1 \leq i \leq k$.

All four of the aforementioned problems are known to be NP-hard, even for small values of k . One way to cope with NP-hardness is to try to design polynomial time approximation algorithms with small approximation ratio. However, apart from DIRECTED MULTIWAY CUT, which admits a 2-approximation in polynomial time [35], all the other three problems are known to have strong lower bounds (functions of n) on the approximation ratio of polynomial time algorithms [16, 19, 25]. Another way to cope with NP-hardness is to try to design FPT algorithms. However, apart from DIRECTED MULTIWAY CUT which has an FPT algorithm parameterized by the size p of the cutset, all the other three problems are known to be W[1]-hard (and hence fixed-parameter intractable) parameterized by size p of the solution X plus the number k of terminals/terminal pairs. When neither of the paradigms of polynomial time approximation algorithms nor (exact) FPT algorithm seem to be successful, the next natural alternative is to try to design FPT approximation algorithms or show hardness of FPT approximation results.

In this paper, we consider the remaining three problems of DIRECTED MULTICUT, STRONGLY CONNECTED STEINER SUBGRAPH and DIRECTED STEINER NETWORK, for which strong approximation and parameterized lower bounds exist, from the viewpoint of FPT approximation algorithms. We obtain two types of results for these three problems: the first is to circumvent the $W[1]$ -hardness and polynomial-time inapproximability results for these problems by designing FPT approximation algorithms, and the second is to strengthen the existing $W[1]$ -hardness by creating “gap-instances” under stronger hypotheses than $FPT \neq W[1]$ such as (Gap-) Exponential Time Hypothesis (ETH). Throughout, we use k to denote number of terminals or terminal pairs and p to denote size of the solution. First, in Section 1.1, we give a brief overview of the current state-of-the-art results for each the three problems from the lens of polynomial time approximation algorithms, FPT algorithms, and FPT approximation algorithms followed by the formal statements of our results. Then, in Section 1.2 we describe the recent flux of results which have set up the framework of FPT hardness of approximation under (Gap-)ETH, and how we use it obtain our hardness results in this paper.

1.1 Previous work and our results

The Directed Multicut problem

Garg et al. [23] showed that DIRECTED MULTICUT is NP-hard even for $k = 2$. The current best approximation ratio in terms of n is $O(n^{11/23} \cdot \log^{O(1)} n)$ due to Agarwal et al. [1], and it is known that DIRECTED MULTICUT is hard to approximate in polynomial time to within a factor of $2^{\Omega(\log^{1-\epsilon} n)}$ for any constant $\epsilon > 0$, unless $NP \subseteq ZPP$ [16]. There is a simple k -approximation in polynomial time obtained by solving each terminal pair as a separate instance of $\min s \rightsquigarrow t$ cut and then taking the union of all the k cuts. Chekuri and Madan [8] and later Lee [30] showed that this is tight: assuming the Unique Games Conjecture of Khot [28], it is not possible to approximate DIRECTED MULTICUT better than factor k in polynomial time, for any fixed k . On the FPT side, Marx and Razgon [34] showed that DIRECTED MULTICUT is $W[1]$ -hard parameterized by p . For the case of bounded k , Chitnis et al. [14] showed that DIRECTED MULTICUT is FPT parameterized by p when $k = 2$, but Pilipczuk and Wahlstrom [36] showed that the problem remains $W[1]$ -hard parameterized by p when $k = 4$. The status of DIRECTED MULTICUT parameterized by p when $k = 3$ is an outstanding open question. We first obtain the following FPT approximation for DIRECTED MULTICUT parameterized by p , which beats any approximation obtainable when parameterizing by k (even in XP time) according to [8, 30]:

► **Theorem 1.** *The DIRECTED MULTICUT problem admits an $\lceil k/2 \rceil$ -approximation in $2^{O(p^2)} \cdot n^{O(1)}$ time.*

The proof of the above theorem uses the FPT algorithm of Chitnis et al. [14, 12] for DIRECTED MULTIWAY CUT parameterized by p as a subroutine. Note that Theorem 1 gives an FPT 2-approximation for DIRECTED MULTICUT WITH 4 PAIRS. We complement this upper bound with a constant factor lower bound for approximation ratio of any FPT algorithm for DIRECTED MULTICUT WITH 4 PAIRS.

► **Theorem 2.** *Under Gap-ETH, for any $\varepsilon > 0$ and any computable function f , there is no $f(p) \cdot n^{O(1)}$ time algorithm that computes an $(\frac{59}{58} - \varepsilon)$ -approximation for DIRECTED MULTICUT WITH 4 PAIRS.*

We did not optimize the constant $59/58$ in order to keep the analysis simple: we believe it can be easily improved, but our techniques would not take it close to the upper bound of 2.

The Directed Steiner Network (DSN) problem

The DSN problem is known to be NP-hard, and furthermore even computing an $O(2^{\log^{1-\varepsilon} n})$ -approximation is not possible [19] in polynomial time, unless $\text{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$. The best known approximation factors for polynomial time algorithms are $O(n^{2/3+\varepsilon})$ and $O(k^{1/2+\varepsilon})$ [4, 7, 21]. On the FPT side, Feldman and Ruhl [20] designed an $n^{O(k)}$ algorithm for DSN (cf. [22]). Chitnis et al. [15] showed that the Feldman-Ruhl algorithm is tight: under ETH, there is no $f(k) \cdot n^{o(k)}$ algorithm (for any computable function f) for DSN even if the input graph is a planar directed acyclic graph. Guo et al. [24] showed that DSN remains W[1]-hard even when parameterized by the larger parameter $k + p$. Dinur and Manurangsi [18] further showed that DSN on general graphs has no FPT approximation algorithm with ratio $k^{1/4-o(1)}$ when parameterized by k , under Gap-ETH.

Chitnis et al. [11] considered two relaxations of the DIRECTED STEINER NETWORK problem: the BI-DSN problem where the input graph is bidirected¹, and the $\text{DSN}_{\text{PLANAR}}$ problem where the input graph is general but the goal is to find a solution whose cost is at most that of an optimal planar solution (if one exists). The main result of Chitnis et al. [11] is that although $\text{BI-DSN}_{\text{PLANAR}}$ (i.e., the intersection of BI-DSN and $\text{DSN}_{\text{PLANAR}}$) is W[1]-hard parameterized by $k + p$, it admits a *parameterized approximation scheme*: for any $\varepsilon > 0$, there is a $\max\{2^{k^{2^{O(1/\varepsilon)}}}, n^{2^{O(1/\varepsilon)}}\}$ time algorithm for $\text{BI-DSN}_{\text{PLANAR}}$ which computes a $(1 + \varepsilon)$ -approximation. Such a parameterized approximation is not possible for BI-DSN as Chitnis et al. [11] showed that under Gap-ETH there is a constant $\alpha > 0$ such that there is no FPT α -approximation. They asked whether a parameterized approximation scheme for the remaining variant of DSN, i.e., the $\text{DSN}_{\text{PLANAR}}$ problem, exists. We answer this question in the negative with the following lower bound

► **Theorem 3.** *Under Gap-ETH, for any $\varepsilon > 0$ and any computable function f , there is no $f(k) \cdot n^{O(1)}$ time algorithm that computes a $(2 - \varepsilon)$ -approximation for $\text{DSN}_{\text{PLANAR}}$, even if the input graph is a directed acyclic graph (DAG).*

The W[1]-hardness proof of [15] for DSN on planar graphs parameterized by k does not give hardness parameterized by p since in that reduction the value of p grows with n . Our next result shows that the slightly more general problem of $\text{DSN}_{\text{PLANAR}}$ (here the input graph is general, but we want to find a solution of cost $\leq p$ if there is a planar solution of size $\leq p$) is indeed W[1]-hard parameterized by $k + p$. Also we obtain a lower bound for approximation schemes for this problem under ETH, i.e., under a weaker assumption than the one used for Theorem 3.²

► **Theorem 4.** *The $\text{DSN}_{\text{PLANAR}}$ problem is W[1]-hard parameterized by $p + k$, even if the input graph is a directed acyclic graph (DAG). Moreover, under ETH, for any computable function f*

- *there is no $f(k, p) \cdot n^{o(k + \sqrt{p})}$ time algorithm for $\text{DSN}_{\text{PLANAR}}$, and*
- *there is no $f(k, \varepsilon, p) \cdot n^{o(k + \sqrt{p+1/\varepsilon})}$ time algorithm which computes a $(1 + \varepsilon)$ -approximation for $\text{DSN}_{\text{PLANAR}}$ for every $\varepsilon > 0$.*

Note that just the W[1]-hardness of $\text{DSN}_{\text{PLANAR}}$ parameterized by $k + p$ already follows from [11] who showed that even the special case of $\text{BI-DSN}_{\text{PLANAR}}$ is W[1]-hard parameterized by $k + p$. However, this reduction from [11] was from ℓ -Clique to an instance of $\text{BI-DSN}_{\text{PLANAR}}$ with $k = O(\ell^2)$ and $p = O(\ell^5)$, whereas Theorem 4 gives a reduction from ℓ -Clique to $\text{DSN}_{\text{PLANAR}}$ with $k = O(\ell)$ and $p = O(\ell^2)$. This gives much improved lower bounds on the running times.

¹ Bidirected graphs are directed graphs which have the property that for every edge $u \rightarrow v$ in G the reverse edge $v \rightarrow u$ exists in G as well and moreover has the same weight as $u \rightarrow v$.

² In the following, $o(f(k, p, \varepsilon))$ means any function $g(f(k, p, \varepsilon))$ such that $g(x) \in o(x)$.

The Strongly Connected Steiner Subgraph (SCSS) problem

The SCSS problem is NP-hard, and the best known approximation ratio in polynomial time for SCSS is k^ϵ for any $\epsilon > 0$ [6]. A result of Halperin and Krauthgamer [25] implies SCSS has no $\Omega(\log^{2-\epsilon} n)$ -approximation for any $\epsilon > 0$, unless NP has quasi-polynomial Las Vegas algorithms. On the FPT side, Feldman and Ruhl [20] designed an $n^{O(k)}$ algorithm for SCSS (cf. [22]). Chitnis et al. [15] showed that the Feldman-Ruhl algorithm is almost optimal: under ETH, there is no $f(k) \cdot n^{o(k/\log k)}$ algorithm (for any computable function f) for SCSS. Guo et al. [24] showed that SCSS remains W[1]-hard even when parameterized by the larger parameter $k + p$. Chitnis et al. [11] showed that the SCSS problem restricted to bidirected graphs remains NP-hard, but is FPT parameterized by k . The SCSS problem admits a *square-root phenomenon* on planar graphs: Chitnis et al. [15] showed that SCSS on planar graphs has an $2^{O(k \log k)} \cdot n^{O(\sqrt{k})}$ algorithm, and under ETH there is a tight lower bound of $f(k) \cdot n^{o(\sqrt{k})}$ for any computable function f . The W[1]-hardness proof of [15] for SCSS on planar graphs parameterized by k does not give hardness parameterized by p , since in that reduction the value of p grows with n . Our next result shows that the slightly more general problem of SCSS_{PLANAR} (here the input graph is general, but we want to find a solution of cost $\leq p$ if there is a planar solution of size $\leq p$) is indeed W[1]-hard parameterized by $k + p$. We also obtain a lower bound for approximation schemes for this problem under ETH:

► **Theorem 5.** *The SCSS_{PLANAR} problem is W[1]-hard parameterized by $p + k$. Moreover, under ETH, for any computable function f*

- *there is no $f(k, p) \cdot n^{o(\sqrt{k+p})}$ time algorithm for SCSS_{PLANAR}, and*
- *there is no $f(k, \varepsilon, p) \cdot n^{o(\sqrt{k+p+1/\varepsilon})}$ time algorithm which computes an $(1+\varepsilon)$ -approximation for SCSS_{PLANAR} for every $\varepsilon > 0$.*

To the best of our knowledge, the only known FPT approximation results for SCSS applied to general graphs parameterized by k : a simple FPT 2-approximation due to Chitnis et al. [13], and a matching $(2 - \epsilon)$ -hardness (for any constant $\epsilon > 0$) under Gap-ETH due to Chitnis et al. [11].

1.2 FPT inapproximability results under (Gap-)ETH

A standard hypothesis for showing lower bounds for running times of FPT and exact exponential time algorithms is the Exponential Time Hypothesis (ETH) of Impagliazzo and Paturi [26].

► **Hypothesis 6. Exponential Time Hypothesis (ETH):** *There exists a constant $\delta > 0$ such that no algorithm can decide whether any given 3-CNF formula is satisfiable in time $O(2^{\delta m})$ where m denotes the number of clauses.*

The original conjecture stated the lower bound as exponential in terms of the number of variables not clauses, but the above statement follows from the Sparsification Lemma of [27]. The Exponential Time Hypothesis has been used extensively to show a variety of lower bounds including those for FPT algorithms, exact exponential time algorithms, hardness of polynomial time approximation, and hardness of FPT approximation. We refer the interested reader to [31] for a survey on lower bounds based on ETH.

To show the W[1]-hardness of DSN_{PLANAR} (Theorem 4) and SCSS_{PLANAR} (Theorem 5) parameterized by $k + p$ we design parameterized reductions from ℓ -Clique to these problems such that $\max\{k, p\}$ is upper bounded by a function of ℓ . Furthermore, by choosing ϵ to be small enough such that computing an $(1 + \epsilon)$ -approximation is the same as computing the

8:6 FPT Inapproximability of Directed Cut and Connectivity Problems

optimal solution, we also obtain runtime lower bounds for $(1 + \epsilon)$ -approximations for these two problems by translating the $f(\ell) \cdot n^{o(\ell)}$ lower bound for ℓ -Clique [9] under ETH (for any computable function f).

Recently, a *gap version* of the ETH was proposed:

► **Hypothesis 7. Gap-ETH** [17, 32]: *There exists a constant $\delta > 0$ such that, given a 3CNF formula Φ on n variables, no $2^{o(n)}$ -time algorithm can distinguish between the following two cases correctly with probability at least $2/3$:*

- Φ is satisfiable.
- Every assignment to the variables violates at least a δ -fraction of the clauses of Φ .

It is known [5, 2] that Gap-ETH follows from ETH given other standard conjectures, such as the existence of linear sized PCPs or exponentially-hard locally-computable one-way functions. We refer the interested reader to [17, 5] for a discussion on why Gap-ETH is a plausible assumption. In a breakthrough result, Chalermsook et al. [5] used Gap-ETH to show that the two famous parameterized intractable problems of Clique and Set Cover are completely inapproximable in FPT time parameterized by the size of the solution. In this paper, we obtain two hardness of approximation results (Theorem 2 and Theorem 3) based on Gap-ETH. The starting point of our hardness of approximation results are based on the recent results on parameterized inapproximability of the DENSEST k -SUBGRAPH problem. Recall that, in the DENSEST k -SUBGRAPH (DkS) problem [29], we are given an undirected graph $G = (V, E)$ and an integer k and the goal is to find a subset $S \subseteq V$ of size ℓ that induces as many edges in G as possible. Chalermsook et al. [5] showed that, under randomized Gap-ETH, there is no FPT approximation (parameterized by k) with ratio $k^{o(1)}$. This was improved recently by Dinur and Manurangsi [18] who showed better hardness and under deterministic Gap-ETH. We state their result formally³:

► **Theorem 8** ([18, Theorem 2]). *Under Gap-ETH, for any function $h(\ell) = o(1)$, there is no $f(\ell) \cdot n^{O(1)}$ -time algorithm that, given a graph G on n vertices and an integer k , can distinguish between the following two cases:*

- (YES) G contains at least one ℓ -clique as a subgraph.
- (NO) Every ℓ -subgraph of G contains less than $\ell^{h(\ell)-1} \cdot \binom{\ell}{2}$ edges.

Note that this result is essentially tight: there is a simple $O(\ell)$ approximation since the number of edges induced by a ℓ -vertex subgraph is at most $\binom{\ell}{2}$ and at least $\lfloor \ell/2 \rfloor$ (without loss of generality, we can assume there are no isolated vertices). Instead of working with DkS, we will reduce from a “colored” version of the problem called MAXIMUM COLORED SUBGRAPH ISOMORPHISM, which can be defined as follows.

Maximum Colored Subgraph Isomorphism (MCSI)

Input : An instance Γ of MCSI consists of three components:

- An undirected graph $G = (V_G, E_G)$,
- A partition of vertex set V_G into disjoint subsets V_1, \dots, V_ℓ ,
- An undirected graph $H = (V_H = \{1, \dots, \ell\}, E_H)$.

Goal: Find an assignment $\phi : V_H \rightarrow V_G$ where $\phi(i) \in V_i$ for every $i \in [\ell]$ that maximizes the number of edges $i - j \in E_H$ such that $\phi(i) - \phi(j) \in E_G$.

³ Dinur and Manurangsi [18] actually state their result for 2-CSPs

This problem is referred to as LABEL COVER in the hardness of approximation literature [3]. However, Chitnis et al. [11] used the name MAXIMUM COLORED SUBGRAPH ISOMORPHISM to be consistent with the naming conventions in the FPT community: this problem is an optimization version of COLORED SUBGRAPH ISOMORPHISM [33]. The graph H is sometimes referred to as the *supergraph* of Γ . Similarly, the vertices and edges of H are called *supernodes* and *superedges* of Γ . Moreover, the size of Γ is defined as $n = |V_G|$, the number of vertices of G . Additionally, for each assignment ϕ , we define its value $\text{val}(\phi)$ to be the fraction of superedges $i - j \in E_H$ such that $\phi(i) - \phi(j) \in E_G$; such superedges are said to be *covered* by ϕ . The objective of MCSI is now to find an assignment ϕ with maximum value. We denote the value of the optimal assignment by $\text{val}(\Gamma)$, i.e., $\text{val}(\Gamma) = \max_{\phi} \text{val}(\phi)$.

Using Theorem 8 we derive the following two corollaries regarding hardness of approximation for MAXIMUM COLORED SUBGRAPH ISOMORPHISM when the supergraph H has special structure. These corollaries follow quite straightforwardly from Theorem 8 using the idea of splitters, but we provide proofs in the full version [10] for completeness.

► **Corollary 9.** [\star]⁴ *Assuming Gap-ETH, for any function $h(\ell) = o(1)$, there is no $f(\ell) \cdot n^{O(1)}$ -time algorithm that, given a MCSI instance Γ of size n such that the supergraph $H = K_{\ell}$, can distinguish between the following two cases:*

- (YES) $\text{val}(\Gamma) = 1$.
- (NO) $\text{val}(\Gamma) < \ell^{h(\ell)-1}$

► **Corollary 10.** [\star] *Assuming Gap-ETH, for any function $h(\ell) = o(1)$, there is no $f(\ell) \cdot n^{O(1)}$ -time algorithm that, given a MCSI instance Γ of size n such that the supergraph H is the complete bipartite subgraph $K_{\frac{\ell}{2}, \frac{\ell}{2}}$, can distinguish between the following two cases:*

- (YES) $\text{val}(\Gamma) = 1$.
- (NO) $\text{val}(\Gamma) < \ell^{h(\ell)-1}$.

We prove Theorem 2 and Theorem 3 via reductions from Corollary 9 and Corollary 10 respectively.

2 FPT (In)Approximability of DIRECTED MULTICUT

In this section we design an FPT 2-approximation for DIRECTED MULTICUT WITH 4 PAIRS parameterized by p (Section 2.1) and complement this with a lower bound (Section 2.2) showing that no FPT algorithm (parameterized by p) for DIRECTED MULTICUT WITH 4 PAIRS can achieve a ratio of $(\frac{59}{58} - \epsilon)$ under Gap-ETH.

2.1 FPT approximation algorithm

It is well-known that a k -approximation can be computed in polynomial time by taking union of min cuts of each of the k terminal pairs. Chekuri and Madan [8] and later Lee [30] showed that this approximation ratio is best-possible for polynomial time algorithms under the Unique Games Conjecture of Khot [28]. The same lower bound also applies for any constant k , i.e., even an XP algorithm parameterized by k cannot compute a better approximation than a polynomial time algorithm. We now design an FPT $\lceil k/2 \rceil$ -approximation for DIRECTED MULTICUT. The idea is borrowed from the proof of Chitnis et al. [14] that DIRECTED MULTICUT WITH 2 PAIRS is FPT parameterized by p .

⁴ All proofs labelled with [\star] appear in the full version [10]

► **Theorem 1.** *The DIRECTED MULTICUT problem admits a $\lceil k/2 \rceil$ -approximation in $2^{O(p^2)} \cdot n^{O(1)}$ time. Formally, the algorithm takes an instance (G, \mathcal{T}) of DIRECTED MULTICUT and in $2^{O(p^2)} \cdot n^{O(1)}$ time either concludes that there is no solution of cost at most p , or produces a solution of cost at most $p \lceil k/2 \rceil$.*

Proof. Let the pairs be $\mathcal{T} = \{(s_i, t_i) : 1 \leq i \leq k\}$, and let OPT be the optimum value for the instance (G, \mathcal{T}) of DIRECTED MULTICUT. For now, assume that k is even. Introduce $k/2$ new vertices r_j, q_j , for $1 \leq j \leq k/2$, of weight $p + 1$ each, and add the following edges:

- $r_j \rightarrow s_{2j-1}$ and $t_{2j-1} \rightarrow q_j$
- $q_j \rightarrow s_{2j}$ and $t_{2j} \rightarrow r_j$

Let the resulting graph be G' , and note that G has an $s_i \rightarrow t_i$ path for some $1 \leq i \leq k$ if and only if G' has a $q_{i/2} \rightarrow r_{i/2}$ or $r_{(i-1)/2} \rightarrow q_{(i-1)/2}$ path (depending on whether i is even or odd). Since the vertices r_j, q_j have weight $p + 1$ each, it follows that G has a solution of size at most p for the instance $(G, \{(s_{2j-1}, t_{2j-1}), (s_{2j}, t_{2j})\})$ of DIRECTED MULTICUT if and only if G' has a solution of size at most p for the DIRECTED MULTIWAY CUT instance with input graph G and terminals r_j, q_j . We use the algorithm of Chitnis et al. [14, 12] for DIRECTED MULTIWAY CUT which checks in $2^{O(p^2)} \cdot n^{O(1)}$ time⁵ if there is a solution of cost at most p . If there is no solution of cost at most p between r_j and q_j in G' then this implies that G has no cut of size at most p separating (s_{2j-1}, t_{2j-1}) and (s_{2j}, t_{2j}) and hence $\text{OPT} > p$. Otherwise, there is a cut C_j in G of cost at most p which separates (s_{2j-1}, t_{2j-1}) and (s_{2j}, t_{2j}) .

The output of the algorithm is the cut $C = \bigcup_{j=1}^{k/2} C_j$. Clearly, if k is even then C is a feasible solution for the instance (G, \mathcal{T}) of DIRECTED MULTICUT with cost at most $\sum_{j=1}^{k/2} \text{cost}(C_j) \leq pk/2$. In case k is odd we use the above procedure for the terminal pairs $\{(s_i, t_i) : 1 \leq i \leq k-1\}$, and finally add a min cut between the last terminal pair (s_k, t_k) . This results in the desired $\lceil k/2 \rceil$ -approximation. ◀

2.2 No FPT $(\frac{59}{58} - \epsilon)$ -approximation under Gap-ETH

With the parameterized hardness of approximating MCSI ready, we can now prove our hardness results for DIRECTED MULTICUT with 4 terminal pairs.

► **Theorem 2.** *Under Gap-ETH, for any $\epsilon > 0$ and any computable function f , there is no $f(p) \cdot n^{O(1)}$ time algorithm that computes an $(\frac{59}{58} - \epsilon)$ -approximation for DIRECTED MULTICUT WITH 4 PAIRS.*

Our proof of the parameterized inapproximability of DIRECTED MULTICUT WITH 4 PAIRS is based on a reduction from MAXIMUM COLORED SUBGRAPH ISOMORPHISM whose properties are described below.

► **Lemma 11.** *There exists a polynomial time reduction that, given an instance $\Gamma = (G, K_\ell, V_1 \cup \dots \cup V_\ell)$ of MCSI, produces an instance (G', \mathcal{T}') of DIRECTED MULTICUT WITH 4 PAIRS such that*

- **(Completeness):** *If $\text{val}(\Gamma) = 1$, then there exists a solution $N \subseteq V(G')$ of cost $29\ell^2$ for the instance (G', \mathcal{T}') of DIRECTED MULTICUT WITH 4 PAIRS*
- **(Soundness):** *If $\text{val}(\Gamma) < \frac{1}{10}$, then every solution $N \subseteq V(G')$ for the instance (G', \mathcal{T}') of DIRECTED MULTICUT WITH 4 PAIRS has cost more than $29.5\ell^2$.*
- **(Parameter Dependency):** *The size of the solution is $p = O(\ell^2)$.*

⁵ This is independent of number of the terminals

In the proof of Lemma 11, we actually use the same reduction as from [36], but with different weights. We reduce to the vertex-weighted variant of DIRECTED MULTICUT WITH 4 PAIRS where we have four different types of weights for the vertices:

- *light* vertices (shown using gray color) which have weight $B = \frac{\ell^2}{\binom{\ell}{2}}$
- *medium* vertices (shown using green color) which have weight $2B$
- *heavy* vertices (shown using orange color) which have weight 20ℓ
- *super-heavy* vertices (shown using white color) which have weight $100\ell^2$

2.2.1 Construction of the Directed Multicut With 4 Pairs instance

Without loss of generality (by adding isolated vertices if necessary) we can assume that $|V_i| = n$ for each $i \in [\ell]$. For each $i \in [\ell]$ let $V_i = \{v_1^i, v_2^i, v_3^i, \dots, v_n^i\}$. Then $|V(G)| = n\ell$. We now describe the construction of the (vertex-weighted) DIRECTED MULTICUT WITH 4 PAIRS instance (G', \mathcal{T}') .

- Introduce eight terminals, arranged in four terminal pairs as follows:

$$\mathcal{T}' = \{(s_{0 \rightarrow n}^x, t_{0 \rightarrow n}^x), (s_{0 \rightarrow n}^y, t_{0 \rightarrow n}^y), (s_{n \rightarrow 0}^<, t_{n \rightarrow 0}^<), (s_{n \rightarrow 0}^>, t_{n \rightarrow 0}^>)\}$$

Each of the 8 terminals is super-heavy.

- For every $1 \leq i \leq \ell$, we introduce a bidirected path on $2n + 1$ vertices (see Figure 2)

$$Z_i := z_0^i \leftrightarrow \hat{z}_1^i \leftrightarrow z_1^i \leftrightarrow \hat{z}_2^i \leftrightarrow z_2^i \leftrightarrow \dots \leftrightarrow \hat{z}_n^i \leftrightarrow z_n^i,$$

called henceforth the *z-path for color class i*. For each $0 \leq a \leq n$ the vertex z_a^i is super-heavy and for each $1 \leq a \leq n$ the vertex \hat{z}_a^i is heavy.

- For every pair (i, j) where $1 \leq i, j \leq \ell$, $i \neq j$, we introduce two bidirected paths (see Figure 2 and Figure 1) on $2n + 1$ vertices

$$X_{i,j} := x_0^{i,j} \leftrightarrow \hat{x}_1^{i,j} \leftrightarrow x_1^{i,j} \leftrightarrow \hat{x}_2^{i,j} \leftrightarrow x_2^{i,j} \leftrightarrow \dots \leftrightarrow \hat{x}_n^{i,j} \leftrightarrow x_n^{i,j}$$

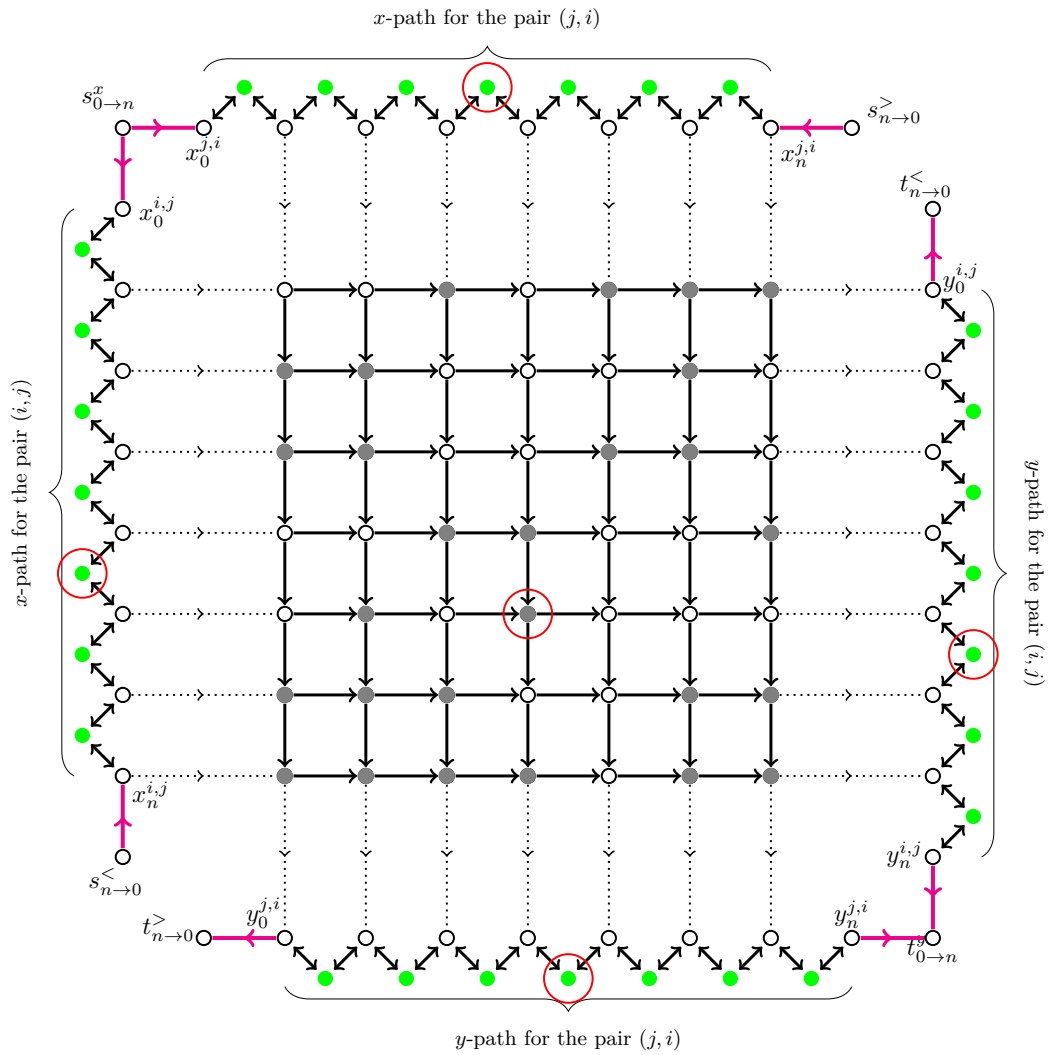
and

$$Y_{i,j} := y_0^{i,j} \leftrightarrow \hat{y}_1^{i,j} \leftrightarrow y_1^{i,j} \leftrightarrow \hat{y}_2^{i,j} \leftrightarrow y_2^{i,j} \leftrightarrow \dots \leftrightarrow \hat{y}_n^{i,j} \leftrightarrow y_n^{i,j}$$

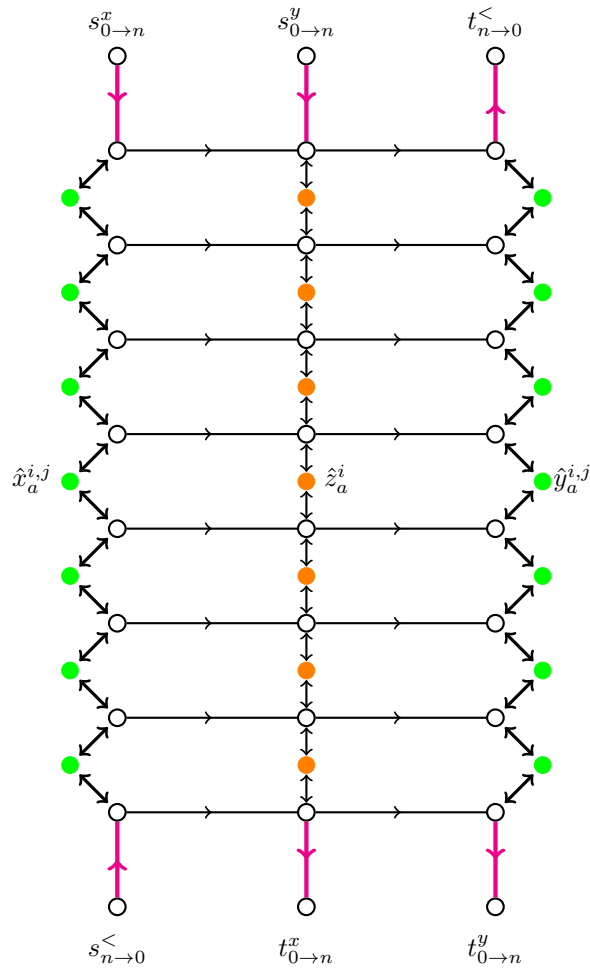
We call these paths the *x-path and the y-path for the pair (i, j)*. For each $0 \leq a \leq n$ the vertices $x_a^{i,j}$ and $y_a^{i,j}$ are super-heavy. For each $1 \leq a \leq n$ the vertices $\hat{x}_a^{i,j}$ and $\hat{y}_a^{i,j}$ are medium.

- For every pair (i, j) with $1 \leq i, j \leq \ell$, $i \neq j$, and every $0 \leq a \leq n$, we add arcs $(x_a^{i,j}, z_a^i)$ and $(z_a^i, y_a^{i,j})$. See Figure 2 for an illustration.
- Furthermore, we attach terminals to the paths as follows: (shown using magenta edges in Figure 1 and Figure 2)
 - for every pair (i, j) with $1 \leq i, j \leq \ell$, $i \neq j$, we add arcs $(s_{0 \rightarrow n}^x, x_0^{i,j})$ and $(y_n^{i,j}, t_{0 \rightarrow n}^y)$;
 - for every $1 \leq i \leq \ell$ we add arcs $(s_{0 \rightarrow n}^y, z_0^i)$ and $(z_n^i, t_{0 \rightarrow n}^x)$;
 - for every pair (i, j) with $1 \leq i < j \leq \ell$ we add arcs $(s_{n \rightarrow 0}^<, x_n^{i,j})$ and $(y_0^{i,j}, t_{n \rightarrow 0}^<)$;
 - for every pair (i, j) with $\ell \geq i > j \geq 1$ we add arcs $(s_{n \rightarrow 0}^>, x_n^{i,j})$ and $(y_0^{i,j}, t_{n \rightarrow 0}^>)$.
- For every pair (i, j) with $1 \leq i < j \leq \ell$ we introduce an acyclic $n \times n$ grid $P_{i,j}$ with vertices $p_{a,b}^{i,j}$ for $1 \leq a, b \leq n$ and arcs $(p_{a,b}^{i,j}, p_{a+1,b}^{i,j})$ for every $1 \leq a < n$ and $1 \leq b \leq n$, as well as $(p_{a,b}^{i,j}, p_{a,b+1}^{i,j})$ for every $1 \leq a \leq n$ and $1 \leq b < n$. We call this grid $P_{i,j}$ as the *p-grid for the pair (i, j)*. We set the vertex $p_{a,b}^{i,j}$ to be a light vertex if $v_a^i v_b^j \in E(G)$, and super-heavy otherwise. Finally, for every $1 \leq a \leq n$ we introduce the following arcs (shown as dotted in Figure 1):

$$(x_a^{i,j}, p_{a,1}^{i,j}), (p_{a,n}^{i,j}, y_{a-1}^{i,j}), (x_a^{j,i}, p_{1,a}^{i,j}), (p_{n,a}^{i,j}, y_{a-1}^{j,i}).$$



■ **Figure 1** Illustration of the reduction for DIRECTED MULTICUT WITH 4 PAIRS. For $1 \leq i < j \leq \ell$, the grid $P_{i,j}$ is surrounded by the bidirectional paths $X_{i,j}$ on the left, $X_{j,i}$ on the top, $Y_{i,j}$ on the right and $Y_{j,i}$ on the bottom. Edges incident on terminals are shown in magenta. Green vertices are medium, orange vertices are heavy and white vertices are super-heavy. A desired solution is marked by red circles.



■ **Figure 2** Illustration of the reduction for DIRECTED MULTICUT WITH 4 PAIRS. For every $1 \leq i < j \leq \ell$, the z -path Z_i corresponding to the color class i is surrounded by the bidirectional paths $X_{i,j}$ on the left and $Y_{i,j}$ on the right. Edges incident on terminals are shown in magenta. Green vertices are medium, orange vertices are heavy and white vertices are super-heavy.

This concludes the construction of the instance (G', \mathcal{T}') of DIRECTED MULTICUT WITH 4 PAIRS. Note that $|V(G')| = (n + \ell)^{O(1)}$, and also G' can be constructed in $(n + \ell)^{O(1)}$ time.

2.2.2 Completeness of Lemma 11:

$$\text{val}(\Gamma) = 1 \Rightarrow \text{Multicut of cost} \leq 29\ell^2$$

Suppose that $\text{val}(\Gamma) = 1$, i.e., G has a ℓ -clique which has exactly one vertex in each V_i for $1 \leq i \leq \ell$. Let this clique be given by $\{v_{\alpha(i)}^i : 1 \leq i \leq \ell\}$. Define

$$X = \{\hat{x}_{\alpha(i)}^{i,j}, \hat{y}_{\alpha(i)}^{i,j} : 1 \leq i, j \leq \ell, i \neq j\} \cup \{\hat{z}_{\alpha(i)}^i : 1 \leq i \leq \ell\} \cup \{p_{\alpha(i), \alpha(j)}^{i,j} : 1 \leq i < j \leq \ell\}.$$

Note that X consists of exactly ℓ heavy $\hat{z}_{\alpha(i)}^i$ vertices, $4\binom{\ell}{2}$ medium $\hat{x}_{\alpha(i)}^{i,j}$ and $\hat{y}_{\alpha(i)}^{i,j}$ vertices, and $\binom{\ell}{2}$ light $p_{\alpha(i), \alpha(j)}^{i,j}$ vertices (the fact that $p_{\alpha(i), \alpha(j)}^{i,j}$ is light for every $1 \leq i < j \leq \ell$ follows from the assumption that the vertices $v_{\alpha(i)}^i$ induce a clique in G). Hence, the weight of X is exactly $\ell \cdot 20\ell + \binom{\ell}{2} \cdot (4 \cdot 2B) + \binom{\ell}{2} \cdot B = 20\ell^2 + \binom{\ell}{2} \cdot 9B = 29\ell^2$. As shown in [36], this set X is a cutset for the instance (G', \mathcal{T}') of DIRECTED MULTICUT WITH 4 PAIRS. The details are deferred to the full version [10].

2.2.3 Soundness of Lemma 11:

$$\text{Multicut of cost} \leq 29.5\ell^2 \Rightarrow \text{val}(\Gamma) \geq \frac{1}{10}$$

Let \mathcal{X} be a solution to the instance (G', \mathcal{T}') of DIRECTED MULTICUT WITH 4 PAIRS such that weight of \mathcal{X} is $29.5\ell^2$. We now show that $\text{val}(\Gamma) \geq \frac{1}{10}$.

► **Observation 12.** *Note that every super-heavy vertex has weight $100\ell^2$ and hence \mathcal{X} cannot contain any super-heavy vertex.*

► **Lemma 13.** $[\star]$ *For each $i \in [\ell]$, the solution \mathcal{X} contains at least one heavy vertex from Z_i .*

► **Lemma 14.** $[\star]$ *For each $1 \leq i \neq j \leq \ell$, the solution \mathcal{X} contains at least one medium vertex from $X_{i,j}$ and at least one medium vertex from $Y_{i,j}$.*

► **Definition 15.** *An integer $i \in [\ell]$ is good if \mathcal{X} contains exactly one heavy vertex from the z -path for the color class i , i.e., $|\mathcal{X} \cap Z_i| = 1$. In this case, we say that $v_{\beta_i}^i$ be the unique vertex from the z -path for class i in the solution \mathcal{X} .*

► **Lemma 16.** $[\star]$ *Let $\text{GOOD} = \{i \in [\ell] : i \text{ is good}\}$. Then $|\text{GOOD}| \geq \frac{37\ell}{40}$.*

► **Definition 17.** *Let $1 \leq i < j \leq \ell$. We say that the pair (i, j) is great if \mathcal{X} contains*

- *exactly one medium vertex from the x -path for the pair (i, j)*
- *exactly one medium vertex from the y -path for the pair (i, j)*
- *exactly one medium vertex from the x -path for the pair (j, i)*
- *exactly one medium vertex from the y -path for the pair (j, i)*
- *exactly one light vertex from the p -grid for the pair (i, j)*

$$\text{Let } \text{GOOD-PAIRS} = \{(i, j) : 1 \leq i < j \leq \ell, i, j \in \text{GOOD}\}$$

► **Lemma 18.** $[\star]$ *Let $1 \leq i < j \leq \ell$. If both i and j are good, and the pair (i, j) is great then $v_{\beta_i}^i - v_{\beta_j}^j \in E(G)$.*

► **Definition 19.** *Let $1 \leq i < j \leq \ell$. We define $\mathcal{X}_{i,j} = \mathcal{X} \cap (X_{i,j} \cup X_{j,i} \cup Y_{i,j} \cup Y_{j,i} \cup P_{i,j})$*

► **Lemma 20.** [★] *Let $1 \leq i < j \leq \ell$ be such that $i, j \in \text{GOOD}$. Then either*

- *the pair (i, j) is great and weight of $\mathcal{X}_{i,j}$ is exactly $9B$, or*
- *weight of $\mathcal{X}_{i,j}$ is at least $10B$*

► **Lemma 21.** [★] *Let $\mathcal{E} = \{1 \leq i < j \leq \ell : i, j \in \text{GOOD} \text{ and } (i, j) \text{ is great}\}$. Then $|\mathcal{E}| \geq \frac{1}{10} \cdot \binom{\ell}{2}$*

Consider the following ℓ -vertex subgraph C : for each $i \in [\ell]$

- if $i \in [\ell]$ is good then add $v_{\beta_i}^i$ to C ,
- otherwise add any vertex from V_i into C .

From Lemma 21 it follows that there are at least $\frac{1}{10} \cdot \binom{\ell}{2}$ edges in G which have both endpoints in C , and hence $\text{val}(\Gamma) \geq \frac{1}{10}$

2.3 Finishing the proof of Theorem 2

We again prove by contrapositive. Suppose that, for some constant $\varepsilon > 0$ and for some computable function $f(p)$ independent of n , there exists an $f(p) \cdot n^{O(1)}$ -time $(\frac{59}{58} - \varepsilon)$ -approximation algorithm for DIRECTED MULTICUT. Let us call this algorithm \mathbb{A} .

We create an algorithm \mathbb{B} that can distinguish between the two cases of Corollary 9 with $h(\ell) = 1 - \frac{\log(10)}{\log \ell} = o(1)$. Our new algorithm \mathbb{B} works as follows. Given an instance $(G, H, V_1 \cup \dots \cup V_\ell)$ of MCSI where $H = K_\ell$, the algorithm \mathbb{B} uses the reduction from Lemma 11 to create a DIRECTED MULTICUT WITH 4 PAIRS instance (G', \mathcal{T}') with 4 terminal pairs. \mathbb{B} then runs \mathbb{A} on this instance with $p = 29\ell^2$; if \mathbb{A} returns a solution N of cost less than $29.5\ell^2$, then \mathbb{B} returns YES. Otherwise, \mathbb{B} returns NO.

To see that algorithm \mathbb{B} can indeed distinguish between the YES and NO cases, first observe that, in the YES case the completeness property of Lemma 11 guarantees that the optimal solution has cost at most $29\ell^2$. Since \mathbb{A} is a $(\frac{59}{58} - \varepsilon)$ -approximation algorithm, it returns a solution of cost at most $(\frac{59}{58} - \varepsilon) \cdot 29\ell^2 < 29.5\ell^2$: this means that \mathbb{B} outputs YES. On the other hand, if $(G, H, V_1 \cup \dots \cup V_\ell)$ is a NO instance, i.e., $\text{val}(\Gamma) < \frac{1}{10} = \ell^{h(\ell)-1}$, then the soundness property of Lemma 22 guarantees that the optimal solution in G' has cost more than $29.5\ell^2$ (which is greater than $(\frac{59}{58} - \varepsilon) \cdot 29\ell^2$) and hence \mathbb{B} correctly outputs NO.

Finally, observe that the running time of \mathbb{B} is $f(p) \cdot |V(G')|^{O(1)}$ plus the $(|V(G)| + \ell)^{O(1)}$ time needed to construct G' . Since $|V(G')| = (|V(G)| + \ell)^{O(1)}$ and $p = O(\ell^2)$ it follows that the total running time is $g(\ell) \cdot |V(G)|$ for some computable function g . Hence, from Corollary 9, Gap-ETH is violated.

3 FPT inapproximability for $\text{DSN}_{\text{Planar}}$

3.1 $(2 - \varepsilon)$ -hardness for FPT approximation under Gap-ETH

The goal of this section is to show the following theorem:

► **Theorem 3.** *Under Gap-ETH, for any $\varepsilon > 0$ and any computable function f , there is no $f(k) \cdot n^{O(1)}$ time algorithm that computes a $(2 - \varepsilon)$ -approximation for $\text{DSN}_{\text{Planar}}$.*

3.1.1 Reduction from Colored Biclique to $\text{DSN}_{\text{Planar}}$

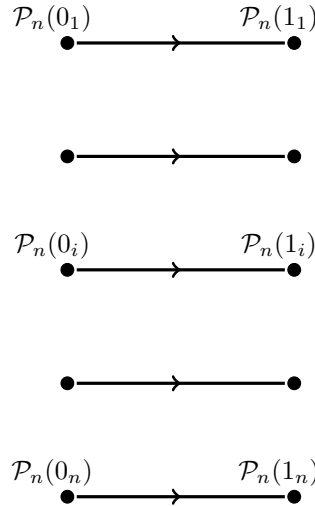
► **Lemma 22.** *For every constant $\gamma > 0$, there exists a polynomial time reduction that, given an instance $\Gamma = (G, H, V_1 \cup \dots \cup V_\ell, W_1, W_2, \dots, W_\ell)$ of MCSI where the supergraph H is $K_{\ell,\ell}$, produces an instance (G', \mathcal{D}') of $\text{DSN}_{\text{Planar}}$, such that*

- **(Completeness)** If $\text{val}(\Gamma) = 1$, then there exists a planar network $N \subseteq G'$ of cost $2(1 + \gamma^{1/5})$ that satisfies all demands.
- **(Soundness)** If $\text{val}(\Gamma) < \gamma$, then every network $N \subseteq G'$ that satisfies all demands has cost more than $2(2 - 4\gamma^{1/5})$.
- *(Parameter Dependency)* The number of demand pairs $k = |\mathcal{D}'|$ is 2ℓ .

Lemma 22 is proven as follows: we construct the $\text{DSN}_{\text{PLANAR}}$ instance in Section 3.1.1.2. The proofs of completeness and soundness of the reduction are deferred to the full version [10]. First, we construct a “path gadget” which we use repeatedly in our construction.

3.1.1.1 Constructing a directed “path” gadget

For every integer n we define the following gadget \mathcal{P}_n which contains $2n$ vertices (see Figure 3). Since we need many of these gadgets later on, we will denote vertices of \mathcal{P}_n by $\mathcal{P}_n(v)$ etc., in order to be able to distinguish vertices of different gadgets. All edges will have the same weight B , which we will fix later during the reductions. The gadget \mathcal{P}_n is constructed as follows: \mathcal{P}_n has a directed path of one edge corresponding to each $i \in [n]$. This is given by $\mathcal{P}_n(0_i) \rightarrow \mathcal{P}_n(1_i)$



■ **Figure 3** The construction of the path gadget for \mathcal{P}_n . Note that the gadget has $2n$ vertices. Each edge of \mathcal{P}_n has the same weight B .

3.1.1.2 Construction of the $\text{DSN}_{\text{PLANAR}}$ instance

We give a reduction which transforms an instance $G = (V, E)$ of $\text{MCSI}(K_{\ell, \ell})$ into an instance of DSN which has 2ℓ demand pairs and an optimum which is planar. Let the partition of V into color classes be given by $\{V_1, V_2, \dots, V_\ell, W_1, W_2, \dots, W_\ell\}$. Without loss of generality (by adding isolated vertices if necessary), we can assume that each color class has the same number of vertices. Let $|V_i| = |W_i| = n'$ for each $1 \leq i \leq \ell$. Then $n = |V(G)| = 2n'\ell$. For each $1 \leq i, j \leq \ell$ we denote by $E_{i,j}$ the set of edges with one end-point in V_i and other in W_j .

We design two types of gadgets: the *main gadget* and the *secondary gadget*. The reduction from $\text{MCSI}(K_{\ell, \ell})$ represents each edge set $E_{i,j}$ with a main gadget $M_{i,j}$. This is done as follows: each main gadget is a copy of the path gadget $\mathcal{P}_{|E_{i,j}|}$ from Section 3.1.1.1 with $B = \frac{2}{\ell^2}$, i.e., there is a row in $M_{i,j}$ corresponding to each edge in $E_{i,j}$. Each main gadget

is surrounded by four secondary gadgets: on the top, right, bottom and left. Each of these gadgets are copies of the path gadget from Section 3.1.1.1 with $B = 0$:

- For each $1 \leq i \leq \ell + 1, 1 \leq j \leq \ell$ the *horizontal* gadget $HS_{i,j}$ is a copy of $\mathcal{P}_{|W_j|}$
- For each $1 \leq i \leq \ell, 1 \leq j \leq \ell + 1$ the *vertical* gadget $VS_{i,j}$ is a copy of $\mathcal{P}_{|V_i|}$

We refer to Figure 4 (bird's-eye view) and Figure 5 (zoomed-in view) for an illustration of the reduction. Fix some $1 \leq i, j \leq \ell$. The main gadget $M_{i,j}$ has four secondary gadgets surrounding it:

- Above $M_{i,j}$ is the vertical secondary gadget $VS_{i,j+1}$
- On the right of $M_{i,j}$ is the horizontal secondary gadget $HS_{i+1,j}$
- Below $M_{i,j}$ is the vertical secondary gadget $VS_{i,j}$
- On the left of $M_{i,j}$ is the horizontal secondary gadget $HS_{i,j}$

Hence, there are $\ell(\ell + 1)$ horizontal secondary gadgets and $\ell(\ell + 1)$ vertical secondary gadgets.

Red intra-gadget edges: Fix (i, j) such that $1 \leq i, j \leq \ell$. Recall that $M_{i,j}$ is a copy of $\mathcal{P}_{|E_{i,j}|}$ with $B = \frac{2}{\ell^2}$ and each of the secondary gadgets are copies of $\mathcal{P}_{n'}$ with $B = 0$. With slight abuse of notation, we assume that the rows of $M_{i,j}$ are indexed by the set $\{(x, y) : (x, y) \in E_{i,j}, x \in W_i, y \in V_j\}$. We add the following edges (in **red** color) of weight 0: for each $(x, y) \in E_{i,j}$

- Add the edge $VS_{i,j+1}(1_x) \rightarrow M_{i,j}(0_{(x,y)})$. These edges are called *top-red* edges incident on $M_{i,j}$.
- Add the edge $HS_{i,j}(1_y) \rightarrow M_{i,j}(0_{(x,y)})$. These edges are called *left-red* edges incident on $M_{i,j}$.
- Add the edge $M_{i,j}(1_{(x,y)}) \rightarrow HS_{i+1,j}(0_y)$. These edges are called *right-red* edges incident on $M_{i,j}$.
- Add the edge $M_{i,j}(1_{(x,y)}) \rightarrow VS_{i,j}(0_x)$. These edges are called *bottom-red* edges incident on $M_{i,j}$.

These are called the *intra-gadget* edges incident on $M_{i,j}$.

Introduce the following 4ℓ vertices (which we call *border* vertices):

- a_1, a_2, \dots, a_ℓ
- b_1, b_2, \dots, b_ℓ
- c_1, c_2, \dots, c_ℓ
- d_1, d_2, \dots, d_ℓ

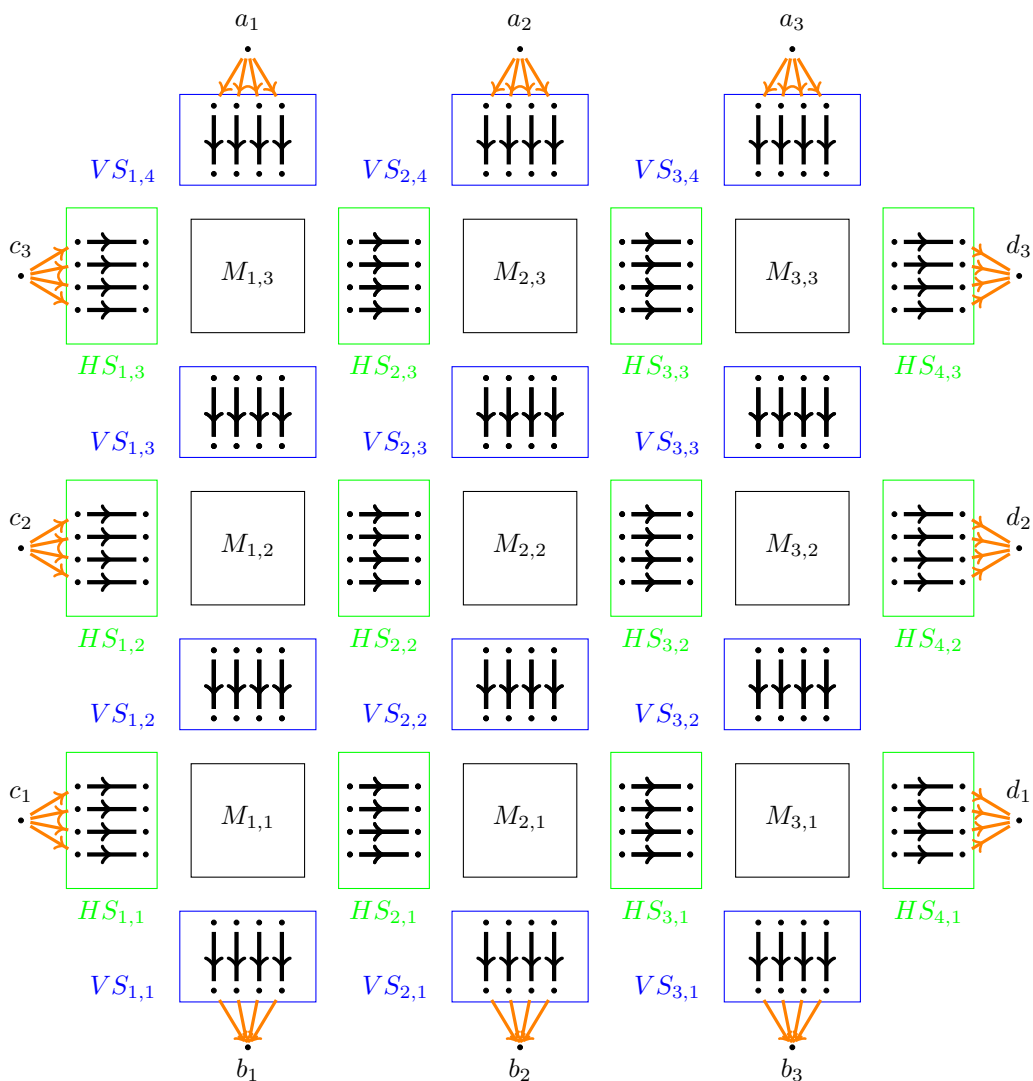
Orange edges: For each $i \in [\ell]$ add the following edges (shown as **orange** in Figure 4) with weight $\frac{2\gamma^{1/5}}{4\ell}$:

- $a_i \rightarrow VS_{i,\ell+1}(0_v)$ for each $v \in V_i$. These are called *top-orange* edges.
- $VS_{i,1}(1_v) \rightarrow b_i$ for each $v \in V_i$. These are called *bottom-orange* edges.
- $c_j \rightarrow HS_{1,j}(0_w)$ for each $w \in W_j$. These are called *left-orange* edges.
- $HS_{\ell+1,j}(1_w) \rightarrow d_j$ for each $w \in W_j$. These are called *right-orange* edges.

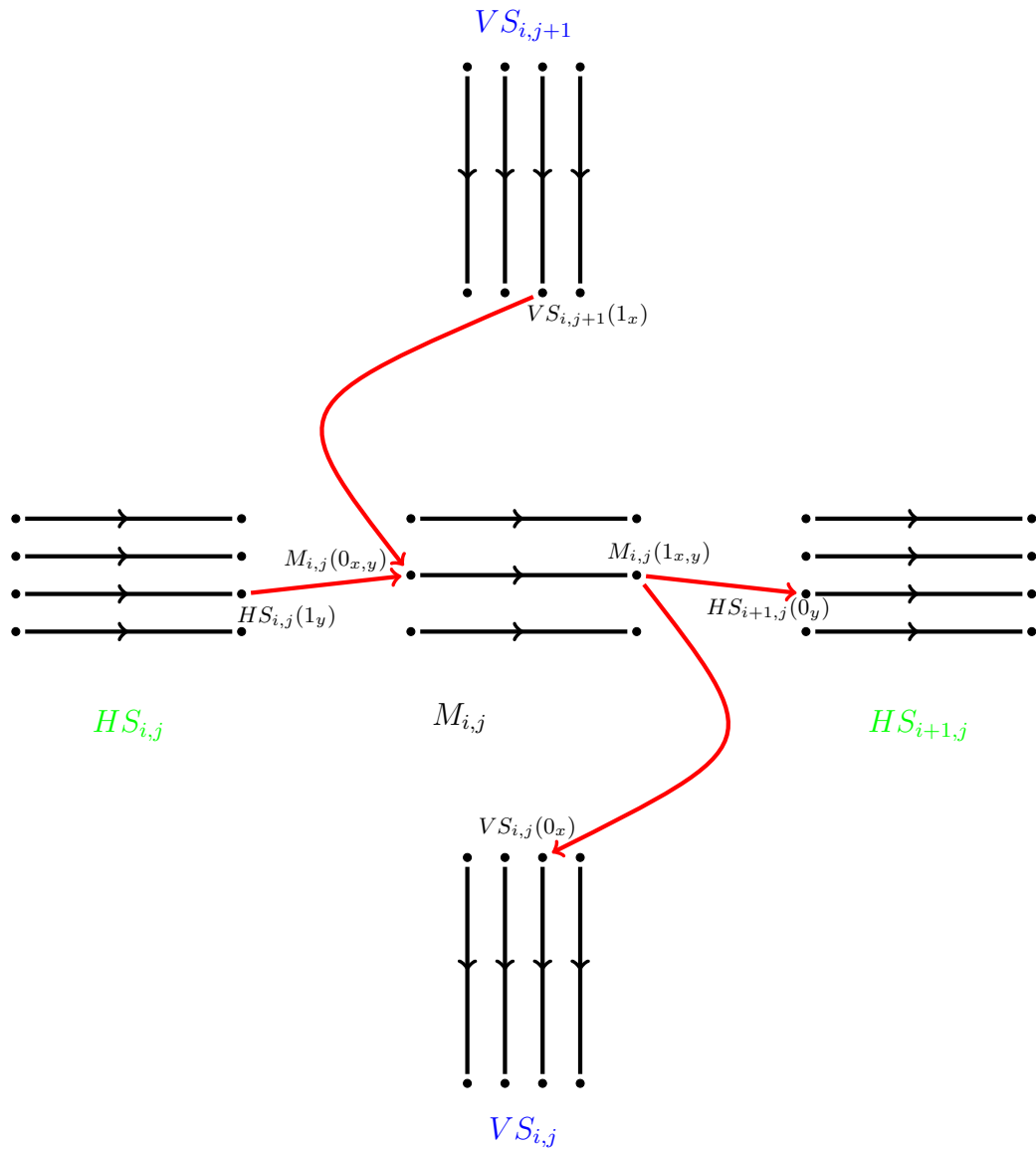
Finally, the set of demand pairs \mathcal{D}' is given by:

- Type I: the pairs (a_i, b_i) for each $1 \leq i \leq \ell$.
- Type II: the pairs (c_j, d_j) for each $1 \leq j \leq \ell$.

Clearly, the total number of demand pairs is $k = |\mathcal{D}'| = 2\ell$. Let the final graph constructed be G' . Note that G' has size $N = (n + \ell)^{O(1)}$ and can be constructed in $(n + \ell)^{O(1)}$ time. It is also easy to see that G' is actually a DAG.



■ **Figure 4** A bird's-eye view of the instance of G' with $\ell = 3$ and $n' = 4$ (see Figure 5 for a zoomed-in view). Additionally we have some red edges between each main gadget and the four secondary gadgets surrounding it which are omitted in this figure for clarity (they are shown in Figure 5 which gives a more zoomed-in view).



■ **Figure 5** A zoomed-in view of the main gadget $M_{i,j}$ surrounded by four secondary gadgets: vertical gadget $VS_{i,j+1}$ on the top, horizontal gadget $HS_{i,j}$ on the left, vertical gadget $VS_{i,j}$ on the bottom and horizontal gadget $HS_{i+1,j}$ on the right. Each of the secondary gadgets is a copy of the uniqueness gadget U_n (see Section 3.1.1.1) and the main gadget $M_{i,j}$ is a copy of the uniqueness gadget $U_{|S_{i,j}|}$. The only inter-gadget edges are the red edges: they have one end-point in a main gadget and the other end-point in a secondary gadget. We have shown four such red edges which are introduced for every $(x, y) \in E_{i,j}$.

3.1.2 Finishing the proof of Theorem 3

We can now easily prove Theorem 3 by combining Lemma 22 and Corollary 10.

Proof of Theorem 3. We again prove by contrapositive. Suppose that, for some constant $\varepsilon > 0$ and for some function $f(k)$ independent of n , there exists an $f(k) \cdot N^{O(1)}$ -time $(2 - \varepsilon)$ -approximation algorithm for $\text{DSN}_{\text{PLANAR}}$ where k is the number of terminal pairs and N is the size of the instance. Let us call this algorithm \mathbb{A} .

Given $\epsilon > 0$, it is easy to see that there exists a sufficiently small $\gamma^* = \gamma^*(\epsilon)$ such that $\frac{2(2-4\gamma^{*1/5})}{2(1+\gamma^{*1/5})} \geq (2 - \epsilon)$. We create an algorithm \mathbb{B} that can distinguish between the two cases of Corollary 10 with $h(\ell) = 1 - \frac{\log(1/\gamma^*)}{\log \ell} = o(1)$. Our new algorithm \mathbb{B} works as follows. Given an instance $(G, H, V_1 \cup \dots \cup V_\ell, W_1 \cup \dots \cup W_\ell)$ of MCSI of size n where $H = K_{\ell, \ell}$, the algorithm \mathbb{B} uses the reduction from Lemma 22 to create in $(n + \ell)^{O(1)}$ time a $\text{DSN}_{\text{PLANAR}}$ instance on the graph G' with $k = 2\ell$ terminal pairs and size $N = (\ell + n)^{O(1)}$. The algorithm \mathbb{B} then runs \mathbb{A} on this instance; if \mathbb{A} returns a solution N of cost at most $2(2 - 4\gamma^{*1/5})$, then \mathbb{B} returns YES. Otherwise, \mathbb{B} returns NO.

We now show that the algorithm \mathbb{B} can indeed distinguish between the YES and NO cases of Corollary 10. In the YES case, i.e., $\text{val}(\Gamma) = 1$, the completeness property of Lemma 22 guarantees that the optimal planar solution has cost at most $2(1 + \gamma^{*1/5})$. Since \mathbb{A} is a $(2 - \varepsilon)$ -approximation algorithm, it returns a solution of cost at most $2(1 + \gamma^{*1/5}) \cdot (2 - \varepsilon) \leq 2(2 - 4\gamma^{*1/5})$ where the inequality comes from our choice of γ^* ; this means that \mathbb{B} outputs YES. On the other hand, in the NO case, i.e., $\text{val}(\Gamma) < \gamma$, the soundness property of Lemma 22 guarantees that the optimal solution (and hence the planar optimal solution as well, if it exists) in G' has cost more than $2(2 - 4\gamma^{*1/5})$, which implies that \mathbb{B} outputs NO.

Finally, observe that the running time of \mathbb{B} is $f(k) \cdot N^{O(1)} + \text{poly}(\ell + n)^{O(1)}$ which is bounded by $f'(\ell) \cdot n^{O(1)}$ for some computable function f' since $k = 2\ell$ and $N = (\ell + n)^{O(1)}$. Hence, from Corollary 10, Gap-ETH is violated. \blacktriangleleft

3.2 Lower Bounds for FPT Approximation Schemes for $\text{DSN}_{\text{PLANAR}}$

We obtain the following result regarding the parameterized complexity of $\text{DSN}_{\text{PLANAR}}$ parameterized by $k + p$.

- **Theorem 4.** $[\star]$ *The $\text{DSN}_{\text{PLANAR}}$ problem is W[1]-hard parameterized by $p + k$. Moreover, under ETH, for any computable function f and any $\varepsilon > 0$*
 - *There is no $f(k, p) \cdot n^{o(k + \sqrt{p})}$ time algorithm for $\text{DSN}_{\text{PLANAR}}$, and*
 - *There is no $f(k, \varepsilon, p) \cdot n^{o(k + \sqrt{p+1/\varepsilon})}$ time algorithm which computes a $(1 + \varepsilon)$ -approximation for $\text{DSN}_{\text{PLANAR}}$*

4 Lower Bounds for FPT Approximation Schemes for $\text{SCSS}_{\text{PLANAR}}$

We obtain the following result regarding the parameterized complexity of $\text{DSN}_{\text{PLANAR}}$ parameterized by $k + p$.

- **Theorem 5.** $[\star]$ *The $\text{SCSS}_{\text{PLANAR}}$ problem is W[1]-hard parameterized by $p + k$. Moreover, under ETH, for any computable function f and any $\varepsilon > 0$*
 - *there is no $f(k, p) \cdot n^{o(\sqrt{k+p})}$ time algorithm for $\text{SCSS}_{\text{PLANAR}}$, and*
 - *there is no $f(k, \varepsilon, p) \cdot n^{o(\sqrt{k+p+1/\varepsilon})}$ time algorithm which computes an $(1 + \varepsilon)$ -approximation for $\text{SCSS}_{\text{PLANAR}}$.*

References

- 1 Amit Agarwal, Noga Alon, and Moses Charikar. Improved approximation for directed cut problems. In *STOC*, pages 671–680, 2007. doi:10.1145/1250790.1250888.
- 2 Benny Applebaum. Exponentially-Hard Gap-CSP and Local PRG via Local Hardcore Functions. In *FOCS 2017*, pages 836–847, 2017. doi:10.1109/FOCS.2017.82.
- 3 Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. doi:10.1006/jcss.1997.1472.
- 4 Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Approximation algorithms for spanner problems and Directed Steiner Forest. *Information and Computation*, 222:93–107, 2013.
- 5 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From Gap-ETH to FPT-Inapproximability: Clique, Dominating Set, and More. In *FOCS*, pages 743–754, 2017. doi:10.1109/FOCS.2017.74.
- 6 Moses Charikar, Chandra Chekuri, To-Yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. Approximation Algorithms for Directed Steiner Problems. *J. Algorithms*, 33(1):73–91, 1999. doi:10.1006/jagm.1999.1042.
- 7 Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. Set connectivity problems in undirected graphs and the directed Steiner network problem. *ACM Transactions on Algorithms*, 7(2):18, 2011.
- 8 Chandra Chekuri and Vivek Madan. Approximating Multicut and the Demand Graph. In *SODA*, pages 855–874, 2017. doi:10.1137/1.9781611974782.54.
- 9 Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006.
- 10 Rajesh Chitnis and Andreas Emil Feldmann. FPT Inapproximability of Directed Cut and Connectivity Problems. *CoRR*, abs/1910.01934, 2019. arXiv:1910.01934.
- 11 Rajesh Chitnis, Andreas Emil Feldmann, and Pasin Manurangsi. Parameterized Approximation Algorithms for Bidirected Steiner Network Problems. In *ESA*, pages 20:1–20:16, 2018. doi:10.4230/LIPIcs.ESA.2018.20.
- 12 Rajesh Hemant Chitnis, Marek Cygan, Mohammad Taghi Hajiaghayi, and Dániel Marx. Directed Subset Feedback Vertex Set Is Fixed-Parameter Tractable. *ACM Trans. Algorithms*, 11(4):28:1–28:28, 2015. doi:10.1145/2700209.
- 13 Rajesh Hemant Chitnis, MohammadTaghi Hajiaghayi, and Guy Kortsarz. Fixed-Parameter and Approximation Algorithms: A New Look. In *IPEC 2013*, pages 110–122, 2013. doi:10.1007/978-3-319-03898-8_11.
- 14 Rajesh Hemant Chitnis, MohammadTaghi Hajiaghayi, and Dániel Marx. Fixed-Parameter Tractability of Directed Multiway Cut Parameterized by the Size of the Cutset. *SIAM J. Comput.*, 42(4):1674–1696, 2013. doi:10.1137/12086217X.
- 15 Rajesh Hemant Chitnis, MohammadTaghi Hajiaghayi, and Dániel Marx. Tight Bounds for Planar Strongly Connected Steiner Subgraph with Fixed Number of Terminals (and Extensions). In *SODA*, pages 1782–1801, 2014. doi:10.1137/1.9781611973402.129.
- 16 Julia Chuzhoy and Sanjeev Khanna. Polynomial flow-cut gaps and hardness of directed cut problems. *J. ACM*, 56(2):6:1–6:28, 2009. doi:10.1145/1502793.1502795.
- 17 Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:128, 2016.
- 18 Irit Dinur and Pasin Manurangsi. ETH-Hardness of Approximating 2-CSPs and Directed Steiner Network. In *ITCS*, pages 36:1–36:20, 2018. doi:10.4230/LIPIcs.ITCS.2018.36.
- 19 Yevgeniy Dodis and Sanjeev Khanna. Design networks with bounded pairwise distance. In *STOC 1999*, pages 750–759, 1999.
- 20 Jon Feldman and Matthias Ruhl. The Directed Steiner Network Problem is Tractable for a Constant Number of Terminals. *SIAM J. Comput.*, 36(2):543–561, 2006. doi:10.1137/S0097539704441241.

- 21 Moran Feldman, Guy Kortsarz, and Zeev Nutov. Improved approximation algorithms for Directed Steiner Forest. *J. Comput. Syst. Sci.*, 78(1):279–292, 2012. doi:10.1016/j.jcss.2011.05.009.
- 22 Andreas Emil Feldmann and Dániel Marx. The Complexity Landscape of Fixed-Parameter Directed Steiner Network Problems. *CoRR*, abs/1707.06808, 2017. arXiv:1707.06808.
- 23 Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Multiway Cuts in Directed and Node Weighted Graphs. In *ICALP*, pages 487–498, 1994. doi:10.1007/3-540-58201-0_92.
- 24 Jiong Guo, Rolf Niedermeier, and Ondrej Suchý. Parameterized Complexity of Arc-Weighted Directed Steiner Problems. *SIAM J. Discrete Math.*, 25(2):583–599, 2011.
- 25 Eran Halperin and Robert Krauthgamer. Polylogarithmic inapproximability. *STOC '03*, pages 585–594, 2003. doi:10.1145/780542.780628.
- 26 Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k -SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.
- 27 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.
- 28 Subhash Khot. On the power of unique 2-prover 1-round games. In *STOC*, pages 767–775, 2002. doi:10.1145/509907.510017.
- 29 Guy Kortsarz and David Peleg. On Choosing a Dense Subgraph (Extended Abstract). In *FOCS 1993*, pages 692–701, 1993.
- 30 Euiwoong Lee. Improved Hardness for Cut, Interdiction, and Firefighter Problems. In *ICALP*, pages 92:1–92:14, 2017. doi:10.4230/LIPIcs.ICALP.2017.92.
- 31 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the Exponential Time Hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011. URL: <http://eatcs.org/beatcs/index.php/beatcs/article/view/92>.
- 32 Pasin Manurangsi and Prasad Raghavendra. A Birthday Repetition Theorem and Complexity of Approximating Dense CSPs. In *ICALP*, pages 78:1–78:15, 2017.
- 33 Dániel Marx. Can You Beat Treewidth? *Theory of Computing*, 6(1):85–112, 2010. doi:10.4086/toc.2010.v006a005.
- 34 Dániel Marx and Igor Razgon. Fixed-Parameter Tractability of Multicut Parameterized by the Size of the Cutset. *SIAM J. Comput.*, 43(2):355–388, 2014. doi:10.1137/110855247.
- 35 Joseph Naor and Leonid Zosin. A 2-Approximation Algorithm for the Directed Multiway Cut Problem. *SIAM J. Comput.*, 31(2):477–482, 2001. doi:10.1137/S009753979732147X.
- 36 Marcin Pilipczuk and Magnus Wahlström. Directed Multicut is $W[1]$ -hard, Even for Four Terminal Pairs. *TOCT*, 10(3):13:1–13:18, 2018. doi:10.1145/3201775.