Parameterized Inapproximability for Steiner Orientation by Gap Amplification

Michał Włodarczyk
Eindhoven University of Technology, The Netherlands
m.wlodarczyk@tue.nl

Abstract

In the k-Steiner Orientation problem, we are given a mixed graph, that is, with both directed and undirected edges, and a set of k terminal pairs. The goal is to find an orientation of the undirected edges that maximizes the number of terminal pairs for which there is a path from the source to the sink. The problem is known to be W[1]-hard when parameterized by k and hard to approximate up to some constant for FPT algorithms assuming Gap-ETH. On the other hand, no approximation factor better than $\mathcal{O}(k)$ is known.

We show that k-Steiner Orientation is unlikely to admit an approximation algorithm with any constant factor, even within FPT running time. To obtain this result, we construct a self-reduction via a hashing-based gap amplification technique, which turns out useful even outside of the FPT paradigm. Precisely, we rule out any approximation factor of the form $(\log k)^{o(1)}$ for FPT algorithms (assuming FPT $\neq$ W[1]) and $(\log n)^{o(1)}$ for purely polynomial-time algorithms (assuming that the class W[1] does not admit randomized FPT algorithms). This constitutes a novel inapproximability result for polynomial-time algorithms obtained via tools from the FPT theory. Moreover, we prove k-Steiner Orientation to belong to W[1], which entails W[1]-completeness of $(\log k)^{o(1)}$-approximation for k-Steiner Orientation. This provides an example of a natural approximation task that is complete in a parameterized complexity class.

Finally, we apply our technique to the maximization version of directed multicut – Max (k,p)-Directed Multicut – where we are given a directed graph, k terminals pairs, and a budget p. The goal is to maximize the number of separated terminal pairs by removing p edges. We present a simple proof that the problem admits no FPT approximation with factor $\mathcal{O}(k^{\frac{1}{2}-\varepsilon})$ (assuming FPT $\neq$ W[1]) and no polynomial-time approximation with ratio $\mathcal{O}(|E(G)|^{\frac{1}{2}-\varepsilon})$ (assuming NP $\not\subseteq$ co-RP).

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability

Keywords and phrases approximation algorithms, fixed-parameter tractability, hardness of approximation, gap amplification

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.104

Category Track A: Algorithms, Complexity and Games

Related Version A full version of the paper is available at `https://arxiv.org/abs/1907.06529`.

Funding The main part of the work has been done when the author was a Ph.D. student at the University of Warsaw and it was a part of the project TOTAL, that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 677651). The author was also supported by the Foundation for Polish Science (FNP).

1 Introduction

In the recent years new research directions emerged in the intersection of the two theories aimed at tackling NP-hard problem: parameterized complexity and approximation algorithms. This led to numerous results combining techniques from both toolboxes. The main goal in this area is to obtain an algorithm running in time $f(k) \cdot |I|^{\mathcal{O}(1)}$ for an instance I with parameter k, that finds a solution of value not worse than α (the approximation factor) times the value of the optimal solution. They are particularly interesting for problems that are both W[1]-hard and at the same time cannot be well approximated in polynomial time [1, 10, 13, 22, 30]. On the other hand, some problems remain resistant to approximation even in this paradigm.

Obtaining polynomial-time approximation lower bounds under the assumption of P $\neq$ NP is challenging, because it usually requires to prove NP-hardness of a *gap problem*. In a gap problem one only needs to distinguish instances with the value of optimal solution at least C_1 from those with this value at most C_2. This provides an argument that one cannot obtain any approximation factor better than the gap, i.e., $\frac{C_1}{C_2}$, as long as P $\neq$ NP.

A road to such lower bounds has been paved by the celebrated PCP theorem [4], which gives an alternative characterization of the class NP. The original complicated proof has been simplified by Dinur [16] via the technique of *gap amplification*: an iterated reduction from a gap problem with a small gap to one with a larger gap. When the number of iterations depends on the input size, this allows us to start the chain of reductions from a problem with no constant gap. However, this is only possible when we can guarantee that the size of all created instances does not grow super-polynomially.

The process of showing approximation lower bounds becomes easier with an additional assumption of the *Unique Games Conjecture* [28], which states that a particular gap version of the Unique Games problem is NP-hard. This makes it possible to start a reduction from a problem with an already relatively large gap. The reductions based on Unique Games Conjecture provided numerous tight approximation lower bounds [5, 23, 31].

A parameterized counterpart of the hardness assumption P $\neq$ NP is FPT $\neq$ W[1], which is equivalent to the statement that k-Clique $\notin$ FPT, that is, k-Clique[1] does not admit an algorithm with running time of the form $f(k) \cdot |I|^{\mathcal{O}(1)}$. Similarly to the classical complexity theory, proving hardness of an approximate task relying only on FPT $\neq$ W[1] is difficult but possible. A recent result stating that the gap version of k-Dominating Set is W[1]-hard (for the gap being any computable function $F(k)$) required gap amplification through a distributed PCP theorem [26].

Again, the task becomes easier when working with a stronger hardness assumption: *Gap Exponential Time Hypothesis*[2] (Gap-ETH) states that there exists $\varepsilon > 0$ so that one requires exponential time to distinguish satisfiable 3-CNF-SAT formulas from those where only a fraction of $(1-\varepsilon)$ clauses can be satisfied at once [17, 34]. Gap-ETH is a stronger assumption than FPT $\neq$ W[1], i.e., the first implies the second, and it sometimes turns out more convenient since it already provides hardness for a problem with a gap. There are many recent examples of using Gap-ETH for showing hardness of parameterized approximation [6, 9, 10, 11, 13, 30].

Our contribution is a novel gap amplification technique which exploits the fact that in a parameterized reduction we can afford an exponential blow-up with respect to the parameter. It circumvents the obstacles related to PCP protocols and, together with a hashing-based lemma, allows us to construct relatively simple self-reductions for problems on directed graphs.

---

1 We attach the parameter to the problem name when we refer to a parameterized problem.

2 Gap-ETH is a stronger version of the *Exponential Time Hypothesis* (ETH), according to which one requires exponential time to solve 3-CNF-SAT [25].

Steiner Orientation

In the k-Steiner Orientation problem we are given a mixed graph, that is, with both directed and undirected edges, and a set of k terminal pairs. The goal is to find an orientation of the undirected edges that maximizes the number of terminal pairs for which there is a path from the source to the sink.

Some of the first studies on the k-Steiner Orientation problem (also referred to as the Maximum Graph Orientation problem) were motivated by modeling protein-protein interactions (PPI) [36] and protein-DNA interactions (PDI) [19, 20]. Whereas PPIs interactions could be represented with undirected graphs, PDIs required introducing mixed graphs. Arkin and Hassin [3] showed the problem to be NP-hard, but polynomially solvable for $k = 2$. This result was generalized by Cygan et al. [15], who presented an $n^{\mathcal{O}(k)}$-time algorithm, which implies that the problem belongs to the class XP (see Section 3) when parameterized by k (cf. [18] for different choices of parameterization).

The k-Steiner Orientation problem has been proved to be W[1]-hard by Pilipczuk and Wahlström [38], which makes it unlikely to be solvable in time $f(k)\cdot|I|^{\mathcal{O}(1)}$. The W[1]-hardness proof has been later strengthened to work on planar graphs and to give a stronger running time lower bound based on ETH [11], which is essentially tight with respect to the $n^{\mathcal{O}(k)}$-time algorithm.

The approximation of Steiner Orientation has been mostly studied on undirected graphs, where the problem reduces to optimization over trees by contracting 2-connected components [15]. Medvedovsky et al. [36] presented an $\mathcal{O}(\log n)$-approximation and actually proved that one can always find an orientation satisfying $\Omega(\frac{k}{\log n})$ terminal pairs. The approximation factor has been improved to $\mathcal{O}(\log n/\log\log n)$ [20] and later to $\mathcal{O}(\log k/\log\log k)$ [15] by observing that one can compress an undirected instance to a tree of size $\mathcal{O}(k)$. A lower bound of $\frac{12}{11} - \varepsilon$ (based on P $\neq$ NP) has been obtained via a reduction from Max Directed Cut [36]. Medvedovsky et al. [36] posed a question of tackling the maximization problem on mixed graphs, which was partially addressed by Gamzu et al. [20] who provided a polylogarithmic approximation in the case where the number of undirected components on each source-sink path is bounded by a constant.

The decision problem whether all the terminal pairs can be satisfied is polynomially solvable when restricting input graphs to be undirected [24], which makes the maximization version fixed-parameter tractable, by simply enumerating all subsets of terminals. The maximization version on mixed graphs is far less understood from the FPT perspective. It is unlikely to be exactly solvable since the decision problem is W[1]-hard, but can we approximate it within a reasonable factor? The reduction by Chitnis et al. [11] implies that, assuming Gap-ETH, k-Steiner Orientation cannot be approximated within factor $\frac{20}{19} - \varepsilon$ on mixed graphs, in running time $f(k) \cdot n^{\mathcal{O}(1)}$. Using new techniques introduced in this paper, we are able to provide stronger lower bounds based on a weaker assumption.

Related work

Some examples of the new advancements in parameterized approximations are 1.81-approximation for k-Cut [22] (recently improved to $(\frac{5}{3} + \varepsilon)$ [27]), which beats the factor 2 that is believed to be optimal within polynomial running time [33], or $(1 + \frac{2}{e} + \varepsilon)$-approximation for k-Median [13], all running in time $f(k) \cdot n^{\mathcal{O}(1)}$. For Capacitated k-Median, a constant factor FPT approximation has been obtained [1, 14], whereas the best-known polynomial-time approximation factor is $\mathcal{O}(\log k)$. Another example is an FPT approximation scheme for the planar case of Bidirected Steiner Network, which does not admit a polynomial-time approximation scheme unless P = NP [10].

On the other hand several problems have proven resistant to such improvements. Chalermsook et al. [9] showed that under the assumption of Gap-ETH there can be no parameterized approximations with ratio $o(k)$ for k-Clique or k-Biclique and none with ratio $F(k)$ for k-Dominating Set (for any computable function F). They have also ruled out $k^{o(1)}$-approximation for Densest k-Subgraph. The cited FPT approximation for k-Median has a tight approximation factor assuming Gap-ETH [13].

Subsequently, efforts have been undertaken to weaken the complexity assumptions on which the lower bounds are based. For the k-Dominating Set problem Gap-ETH has been replaced with a more established hardness assumption that FPT $\neq$ W[1] [26]. Marx [35] has proven parameterized inapproximability of Monotone k-Circuit SAT under the even weaker assumption that FPT $\neq$ W[P]. Lokshtanov et al. [30] introduced the *Parameterized Inapproximability Hypothesis* (PIH), that is weaker than Gap-ETH and stronger than FPT $\neq$ W[1], and used it to rule out an FPT approximation scheme for Directed k-Odd Cycle Transversal. PIH turned out to be a sufficient assumption to argue there can be no FPT algorithm for k-Even Set [6].

2 Overview of the results

Our main inapproximability result is a W[1]-hardness proof for the gap version of k-Steiner Orientation with the gap $q = (\log k)^{o(1)}$. This means that the problem is unlikely to admit a $(\log k)^{o(1)}$-approximation algorithm with running time $f(k) \cdot |I|^{\mathcal{O}(1)}$.

▶ **Theorem 2.1.** *Consider a function $\alpha(k) = (\log k)^{\beta(k)}$, where $\beta(k) \to 0$ is computable and non-increasing. It is* W[1]*-hard to distinguish whether for a given instance of k-Steiner Orientation:*

1. *there exists an orientation satisfying all k terminal pairs, or*
2. *for all orientations the number of satisfied pairs is at most $\frac{1}{\alpha(k)} \cdot k$.*

The previously known approximation lower bound for FPT algorithms, $\frac{20}{19} - \varepsilon$, was obtained via a linear reduction from k-Clique and was based on Gap-ETH [11]. Our reduction not only raises the inapproximability bar significantly, but also weakens the hardness assumption (although we are not able to enforce the planarity of the produced instances, as in [11]). In fact, we begin with the decision version of k-Steiner Orientation and introduce a gap inside the self-reduction. What is interesting, we rely on totally different properties of the problem than in the W[1]-hardness proof [38]: that one required gadgets with long undirected paths and we introduce only new directed edges.

This result is also interesting from the perspective of the classical (non-parameterized) approximation theory. The best approximation lower bound known so far has been $\frac{12}{11} - \varepsilon$ [36], valid also for undirected graphs. Therefore we provide a new inapproximability result for polynomial algorithms, which is based on an assumption from parameterized complexity. Restricting to a purely polynomial running time allows us to rule out also approximation factors depending on n (rather than on k) with a slightly stronger assumption, which is required because the reduction is randomized (see Section 3 for the formal definition of a false-biased FPT algorithm).

▶ **Theorem 2.2.** *Consider a function $\alpha(n) = (\log n)^{\beta(n)}$, where $\beta(n) \to 0$ is computable and non-increasing. Unless the class W[1] admits false-biased FPT algorithms, there is no polynomial-time algorithm that, given an instance of Steiner Orientation with n vertices and k terminal pairs, distinguishes between the following cases:*

1. *there exists an orientation satisfying all k terminal pairs, or*
2. *for all orientations the number of satisfied pairs is at most $\frac{1}{\alpha(n)} \cdot k$.*

A similar phenomenon, that is, novel polynomial-time hardness based on an assumption from parameterized complexity, has appeared in the work on MONOTONE k-CIRCUIT SAT [35]. Another example of this kind is polynomial-time approximation hardness for DENSEST k-SUBGRAPH based on ETH [32].

W[1]-completeness

So far, the decision version of k-STEINER ORIENTATION has only been known to be W[1]-hard [38] and to belong to XP [15]. We establish its exact location in the W-hierarchy. A crucial new insight is that we can assume the solution to be composed of $f(k)$ pieces, for which we only need to check if they match each other, and this task reduces to k-CLIQUE.

▶ **Theorem 2.3.** *k-STEINER ORIENTATION is* W[1]*-complete.*

We hereby solve an open problem posted by Chitnis et al. [11]. What is more, this implies that $(\log k)^{o(1)}$-GAP k-STEINER ORIENTATION belongs to W[1] (see Section 3 for formal definitions of problems). Together with Theorem 2.1 this entails W[1]-completeness. Another gap problem with this property is MAXIMUM k-SUBSET INTERSECTION[3], introduced for the purpose of proving W[1]-hardness of k-BICLIQUE [29]. We are not aware of any other natural gap problem being complete in a parameterized complexity class. Note that although W[1]-hardness of the gap version of k-DOMINATING SET is known [26], k-DOMINATING SET is W[2]-complete.

Directed Multicut

As another application of our technique, we present a simple hardness result for the gap version of MAX (k, p)-DIRECTED MULTICUT with the gap $q = k^{\frac{1}{2}-\varepsilon}$. We show that even if we parameterize the problem with both the number of terminal pairs k and the size of the cutset p, then we essentially cannot obtain any approximation ratio better than $\sqrt{k}$.

▶ **Theorem 2.4.** *For any $\varepsilon > 0$ and function $\alpha(k) = \mathcal{O}\left(k^{\frac{1}{2}-\varepsilon}\right)$, it is* W[1]*-hard to distinguish whether for a given instance of* MAX (k, p)-DIRECTED MULTICUT*:*

1. *there is a cut of size p that separates all k terminal pairs, or*
2. *all cuts of size p separate at most $\frac{1}{\alpha(k)} \cdot k$ terminal pairs.*

When restricted to polynomial running time, the lower bound of $\Omega(k^{\frac{1}{2}-\varepsilon})$ can be improved to $\Omega(|E(G)|^{\frac{1}{2}-\varepsilon})$, however unlike the case of k-STEINER ORIENTATION, this time the reduction is polynomial and we need to assume only NP $\not\subseteq$ co-RP (recall that a problem is in co-RP if it admits a polynomial-time false-biased algorithm, i.e., an algorithm which is always correct for YES-instances and for NO-instances returns the correct answer with probability greater than some constant).

▶ **Theorem 2.5.** *Assuming* NP $\not\subseteq$ co-RP*, for any $\varepsilon > 0$ and function $\alpha(m) = \mathcal{O}\left(m^{\frac{1}{2}-\varepsilon}\right)$, there is no polynomial-time algorithm that, given an instance $(G, \mathcal{T}, p)$, $|\mathcal{T}| = k$, $|E(G)| = m$, of* MAX DIRECTED MULTICUT*, distinguishes between the following cases:*

1. *there is a cut of size p that separates all k terminal pairs, or*
2. *all cuts of size p separate at most $\frac{1}{\alpha(m)} \cdot k$ terminal pairs.*

[3] To give a concrete example of a W[1]-complete gap problem, consider the task of distinguishing graphs with $K_{k,F_1(k)}$ from those with no $K_{k,F_2(k)}$. The reduction in [29] implies that this is W[1]-hard for functions F_1, F_2 with large gap. On the other hand, the problem of finding $K_{k,F_1(k)}$ belongs to W[1] via a color-coding reduction to $(F_1(k) + k)$-CLIQUE.

As far as we know, the approximation status of this variant has not been studied yet. If we want to minimize the number of removed edges to separate all terminal pairs or minimize the ratio of the cutset size to the number of separated terminal pairs (this problem is known as Directed Sparsest Multicut), those cases admit a polynomial-time $\tilde{\mathcal{O}}(n^{\frac{11}{23}})$-approximation algorithm [2] and a lower bound of $2^{\Omega(\log^{1-\varepsilon} n)}$ [12]. Since $\frac{11}{23} < \frac{1}{2}$ and $n \leq m$, the maximization variant with a hard constraint on the cutset size turns out to be harder.

In the undirected case p-Multicut is FPT, even when parameterized only by the size of the cutset p and allowing arbitrarily many terminals [7]. This is in contrast with the directed case, which becomes W[1]-hard already for 4 terminals, when parameterized by p. It is worth mentioning that k-Steiner Orientation and p-Directed Multicut were proven to be W[1]-hard with a similar gadgeting machinery [38].

Organization of the paper

We begin with the necessary definitions in Section 3. As our gap amplification technique is arguably the most innovative ingredient of the paper, we precede the proofs with informal Section 4, which introduces the ideas gradually. It is followed by the detailed constructions for k-Steiner Orientation in Section 5 and for Max (k, p)-Directed Multicut in Section 6. Each contains a self-reduction lemma and applications to polynomial and FPT running time. The proof of W[1]-completeness of k-Steiner Orientation can be found in the full version of the article.

3 Preliminaries

Fixed parameter tractability

A parameterized problem instance is created by associating an integer parameter k with an input instance. Formally, a parameterized language is a subset of $\Sigma^* \times \mathbb{N}$. We say that a language (or a problem) is *fixed parameter tractable* (FPT) if it admits an algorithm solving an instance (I, k) (i.e., deciding if it belongs to the language) in running time $f(k) \cdot |I|^{\mathcal{O}(1)}$, where f is a computable function. Such a procedure is called an *FPT algorithm* and we say concisely that it runs in *FPT time*. A language belongs to the broader class XP if it admits an algorithm with running time of the form $|I|^{f(k)}$.

There is no widely recognized class describing problems which admit randomized FPT algorithms. Instead of defining such a class, we will directly use the notion of a *false-biased* algorithm, which is always correct for YES-instances and for NO-instances returns the correct answer with probability greater than some constant (equivalently, when the algorithm returns *false* then it is always correct). Similarly, a *true-biased* algorithm is always correct for NO-instances but may be wrong for YES-instances with bounded probability. A false-biased (resp. true-biased) FPT algorithm satisfies the condition above and runs in FPT time.

To argue that a problem is unlikely to be FPT, we use parameterized reductions analogous to those employed in the classical complexity theory. Here, the concept of W-hardness replaces NP-hardness, and we need not only to construct an equivalent instance in FPT time, but also ensure that the parameter in the new instance depends only on the parameter in the original instance. If there exists a parameterized reduction from a W[1]-hard problem (e.g., k-Clique) to another problem Π, then the problem Π is W[1]-hard as well. This provides an argument that Π does not admit an algorithm with running time $f(k) \cdot |I|^{\mathcal{O}(1)}$ under the assumption that FPT $\neq$ W[1].

Approximation algorithms and gap problems

We define an optimization problem (resp. parameterized optimization problem) as a task of optimizing function $L \to \mathbb{N}$, where $L \subseteq \Sigma^*$ (resp. $L \subseteq \Sigma^* \times \mathbb{N}$), representing the value of the optimal solution. An α-approximation algorithm for a maximization task must return a solution of value no less than the optimum divided by α (we follow the convention with $\alpha > 1$). The approximation factor α can be a constant or it can depend on the input size. In the most common setting, the running time is required to be polynomial. An FPT approximation algorithm works with a parameterized optimization problem and is required to run in FPT time. It is common that its approximation factor can depend on the parameter.

A gap problem (resp. parameterized gap problem) is given by two disjoint languages $L_1, L_2 \subseteq \Sigma^*$ (resp. $L_1, L_2 \subseteq \Sigma^* \times \mathbb{N}$). An algorithm should decide whether the input belongs to L_1 or to L_2. If neither holds, then the algorithm is allowed to return anything. Usually L_1, L_2 are defined respectively as the sets of instances (of an optimization problem) with a solution of value at least C_1 and instances with no solution with value greater than C_2. An α-approximation algorithm with $\alpha < \frac{C_1}{C_2}$ can distinguish L_1 from L_2, therefore hardness of an approximation task is implied by hardness for the related gap problem.

Problem definitions

We now formally describe the problems we work with. Since we consider parameterized algorithms it is important to specify how we define the parameter of an instance.

A mixed graph is a triple (V, A, E), where V is the vertex set, A is the set of directed edges, and E stands for the set of undirected edges. An orientation of a mixed graph is given by replacing each undirected edge $uv \in E$ with one of the directed ones: (u, v) or (v, u). This creates a directed graph $\tilde{G} = (V, A \cup \tilde{E})$, where $\tilde{E}$ is the set of newly created directed edges. We assume that $uv \notin E$ for each $(u, v) \in A$, so $\tilde{G}$ is always a simple graph.

k-Steiner Orientation

Input:	mixed graph $G = (V, A, E)$, list of terminal pairs $\mathcal{T} = ((s_1, t_1), \dots, (s_k, t_k))$,
Parameter:	k
Task:	find an orientation $\tilde{G}$ of G that maximizes the number of pairs (s_i, t_i), such that t_i is reachable from s_i in $\tilde{G}$

We add „Max" to the name of the following problem in order to distinguish it from the more common version of Directed Multicut, where one minimizes the number of edges in the cut.

Max (k, p)-Directed Multicut

Input:	directed graph $G = (V, A)$, list of terminal pairs $\mathcal{T} = ((s_1, t_1), \dots, (s_k, t_k))$, integer p
Parameter:	$k + p$
Task:	find a subset of edges $A' \subseteq A$, $\lvert A' \rvert \le p$, in order to maximize the number of pairs (s_i, t_i), such that t_i is unreachable from s_i in $G \setminus A'$

If a solution to either problem satisfies the reachability (resp. unreachability) condition for a particular terminal pair, we say that this pair is satisfied by this solution. The decision versions of both problems ask whether there is a solution of value k, that is, satisfying all the terminal pairs. We call such an instance fully satisfiable, or a YES-instance, and a NO-instance otherwise. For the sake of proving approximation hardness we introduce the gap versions: q-Gap k-Steiner Orientation and q-Gap Max (k,p)-Directed Multicut, where we are promised that the value of the optimal solution is either k or at most $\frac{k}{q}$, and we have to distinguish between these cases.

When referring to non-parameterized problems, we drop the parameters in the problem name. We use notation $[n] = \{1, 2, \ldots, n\}$. All logarithms are 2-based.

4 The gap amplification technique

We begin with an informal thought experiment that helps to understand the main ideas behind the reduction. For an instance $(G, \mathcal{T} = ((s_1, t_1), \ldots, (s_k, t_k)))$ of k-Steiner Orientation we refer to the vertices $s_i, t_i \in V(G)$ as $G\{s, i\}$ and $G\{t, i\}$. We want to construct a larger instance $(H, \mathcal{T}_H)$ so that if $(G, \mathcal{T})$ is fully satisfiable then $(H, \mathcal{T}_H)$ is as well, but otherwise the maximal fraction of satisfiable pairs in $(H, \mathcal{T}_H)$ is strictly less than $\frac{k-1}{k}$. Consider k vertex-disjoint copies of the original instance: $(G_1, \mathcal{T}_1), (G_2, \mathcal{T}_2), \ldots, (G_k, \mathcal{T}_k)$, that will be treated as the first layer. Assume that $(G, \mathcal{T})$ is a NO-instance (i.e., one cannot satisfy all pairs at once), so for any orientation of the copies $\tilde{G}_1, \tilde{G}_2, \ldots, \tilde{G}_k$, there is a tuple $(j_1, j_2, \ldots, j_k)$ such that $\tilde{G}_i\{t, j_i\}$ is unreachable from $\tilde{G}_i\{s, j_i\}$ in $\tilde{G}_i$. Suppose for now that we have fixed the values of (j_i), even before we have finished building our instance.

Let $R = (r_1, r_2, \ldots, r_k)$ be a tuple sampled randomly from $[k]^k$. We connect the sinks in the first layer to the sources in another copy of the same instance – let us refer to it as $(G_R, \mathcal{T}_R)$. We add a directed edge from $G_i\{t, r_i\}$ to $G_R\{s, i\}$ for each $i \in [k]$, thus connecting a random sink of G_i to the source $G_R\{s, i\}$, as shown in Figure 1. We refer to the union of all $k+1$ copies of G with k added connecting edges as the graph H. We define $\mathcal{T}_H = ((G_1\{s, r_1\}, G_R\{t, 1\}), \ldots, (G_k\{s, r_k\}, G_R\{t, k\}))$, so we want to satisfy those k terminal pairs that got connected randomly. Let X be a random variable equal to the value of the optimal solution for $(H, \mathcal{T}_H)$ under the restriction that the solution orients $G_1, G_2 \ldots, G_k$ as $\tilde{G}_1, \tilde{G}_2, \ldots, \tilde{G}_k$. What would be the expected value of X?

Let Y denote another random variable being the number of indices i for which $r_i \neq j_i$. By linearity of expectation we have $\mathbb{E}Y = k - 1$. It holds that $X \leq Y$ and so far we still have only the bound $\mathbb{E}X \leq k - 1$. However, with probability $(\frac{k-1}{k})^k$ we have $r_i \neq j_i$ for all i, therefore $Y = k$, but we cannot connect all pairs within G_R (because it is a copy of a NO-instance), so $X \leq k - 1$. This means that $\mathbb{E}(Y - X) \geq (\frac{k-1}{k})^k$ and so $\mathbb{E}X \leq k - 1 - (\frac{k-1}{k})^k$: the gap has been slightly amplified.

Of course in the proper reduction we cannot fix the orientation before adding the connecting edges. However, we can afford an exponential blow-up with respect to k. We can include in the second layer the whole probabilistic space, that is, k^k copies of $(G, \mathcal{T})$ (rather than a single $(G_R, \mathcal{T}_R)$), each connected to the first layer with respect to a different tuple $(r_1, r_2, \ldots, r_k)$, thus creating a large instance $(H, \mathcal{T}_H)$ with $k^k \cdot k$ terminal pairs (see Figure 1). For any orientation of H the fraction of satisfied terminal pairs equals the average over the fractions for all k^k groups of terminal pairs, so we can emulate the construction above without fixing $(j_1, j_2, \ldots, j_k)$. The maximal fraction of satisfiable terminal pairs in such (no longer random) $(H, \mathcal{T}_H)$ would be the same as before, that is, $\frac{\mathbb{E}X}{k} < k - 1$. However, the smaller instance we create the better lower bounds we get, so we will try to be more economical while constructing $(H, \mathcal{T}_H)$.

An important observation is that we do not have to include all k^k choices of R in the construction. We just need a sufficient combination of them, so that the gap amplification occurs for any choice of $(j_1, j_2, \ldots, j_k)$. This can be ensured by picking just $k^{\mathcal{O}(1)}$ choices of R and using an argument based on the Chernoff bound (see Section 5 for a detailed construction).

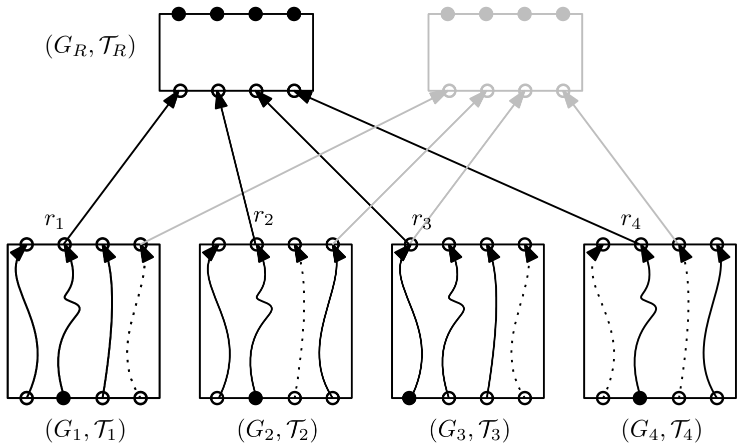

Figure 1 The construction of $(H, \mathcal{T}_H)$ for $k = 4$. The black copy $(G_R, \mathcal{T}_R)$ is connected according to the choice of $R = (r_1, r_2, r_3, r_4) = (2, 2, 1, 2)$ and the terminals (sources and sinks) are marked with black dots. The dotted lines indicate which pairs of terminals are unreachable in each $\tilde{G}_i$ and we can set $(j_1, j_2, j_3, j_4) = (4, 3, 4, 3)$ (another valid choice is $(4, 3, 4, 1)$). We have $r_i \neq j_i$ for each i, so $Y = 4$ but $X \leq 3$. The grey copy illustrates another random choice of connection: $R' = (r_1, r_2, r_3, r_4) = (4, 4, 1, 3)$ and $X \leq Y = 2$. If we wanted to construct $(H, \mathcal{T}_H)$ emulating the whole probabilistic space, we would include all 4^4 copies of $(G_R, \mathcal{T}_R)$ for all choices of $R = (r_1, r_2, r_3, r_4)$ in the same way as the black and the grey copy.

We can iterate this construction by treating $(H, \mathcal{T}_H)$ as a new input and amplifying the gap further in each step. In further steps we need to add an exponential number of copies to the new layer, even when compressing the probabilistic space as above. This is why we get an exponential blow-up with respect to k and we need to work with a parameterized hardness assumption, even for ruling out polynomial-time approximations.

The construction for Max (k, p)-Directed Multicut is simpler because the layer stacking does not have to be iterated. Therefore to achieve polynomial-time hardness it suffices to assume that NP $\not\subseteq$ co-RP. The phenomenon that both problems admit such strong self-reduction properties can be explained by the fact that when dealing with directed reachability one can compose instances sequentially, which is the first step in both reductions.

5 Inapproximability of Steiner Orientation

We are going to present a formal construction of the argument sketched in Section 4. We first formulate and discuss the properties of the construction. Then we introduce a probabilistic tool, called a δ-biased sampler family, describe the reduction step, and prove the gap amplifying property. At the end of this section we present the proof of the following lemma. Let $S(G, \mathcal{T})$ denote the maximal number of pairs that can be satisfied by some orientation in an instance $(G, \mathcal{T})$.

▶ **Lemma 5.1.** *There is a procedure that, for an instance $(G, \mathcal{T}_G)$, $k = |\mathcal{T}_G|$, of Steiner Orientation, and a parameter q, constructs a new instance $(H, \mathcal{T}_H)$, $k_0 = |\mathcal{T}_H|$, such that:*
1. $k_0 = 2^{q^{\mathcal{O}(k)}}$,
2. $|V(H)| \leq |V(G)| \cdot k_0^2$,
3. *if $S(G, \mathcal{T}_G) = k$, then $S(H, \mathcal{T}_H) = k_0$ always (Completeness),*
4. *if $S(G, \mathcal{T}_G) < k$, then $S(H, \mathcal{T}_H) \leq \frac{1}{q} \cdot k_0$ with probability at least $\frac{1}{k_0}$ (Soundness).*

The randomized construction runs in time proportional to $|H|$. It can be derandomized within running time $f(k, q) \cdot |G|$.

It easily follows from these properties that the gap can get amplified to any constant q. It is more complicated though to rule out a superconstant approximation factor, e.g., $\alpha(k) = \log\log k$, because we need to keep track of the growth of $\alpha(k_0)$ when increasing q. We address this issue after proving Lemma 5.1.

Sampler families

As sketched in Section 4, given fixed orientations of the k copies of G, we are able to randomly sample k sinks and insert additional edges so that the expected optimum of the new instance is sufficiently upper bounded. We want to reverse this idea, so we could randomly sample a moderate number of additional connections once to ensure the upper bound works for any orientation. To this end, we need some kind of a hashing technique to mimic the behaviour of the probabilistic space with a structure of moderate size. Examples of such constructions are (generalized) universal hash families [8, 39, 40] or expander random walk sampling [21]. Even though the construction presented below is relatively simple, we are not aware of any occurrences of it in the literature.

For a set X_1 and a multiset X_2, we write $X_2 \subseteq X_1$ if every element from X_2 appears in X_1. Let $U(X)$ denote the uniform distribution over a finite multiset X. In particular, each distinct copy of the same element in X has the same probability of being chosen: $\frac{1}{|X|}$.

▶ **Definition 5.2.** *For a family $\mathcal{F}$ of functions $X \to [0,1]$, a δ-biased sampler family is a multiset $X_H \subseteq X$, such that for every $f \in \mathcal{F}$ it holds*

$$\left|\mathbb{E}_{x\sim U(X_H)} f(x) - \mathbb{E}_{x\sim U(X)} f(x)\right| \leq \delta.$$

▶ **Lemma 5.3.** *For a given X, $\mathcal{F}$, and $\delta > 0$, a sample of $\mathcal{O}(\delta^{-2}\log(|\mathcal{F}|))$ elements from X (sampled independently with repetitions) forms a δ-biased sampler family with probability at least $\frac{1}{2}$.*

Proof. We sample independently $M = 10 \cdot \delta^{-2}\log(|\mathcal{F}|)$ elements from X with repetitions. For the sake of analysis, note that this is a single sample from the space $\Omega_{X,M}$ being the family of all M-tuples of elements from X, equipped with a uniform distribution. Let X_H denote the random multiset of all elements in this M-tuple. For each $f \in \mathcal{F}$ we define $A_f \subseteq \Omega_{X,M}$ as the family of tuples for which $\left|\mathbb{E}_{x\sim U(X_H)} f(x) - \mathbb{E}_{x\sim U(X)} f(x)\right| > \delta$. For a fixed f we apply the Hoeffding's inequality.

$$\mathbb{P}(A_f) = \mathbb{P}\left(\left|\mathbb{E}_{x\sim U(X_H)} f(x) - \mathbb{E}_{x\sim U(X)} f(x)\right| > \delta\right) \leq 2\exp(-2\delta^2 M).$$

For our choice of M this bound gets less than $\frac{1}{2|\mathcal{F}|}$. By union bound, the probability that X_F is not a δ-biased sampler family is $\mathbb{P}\left(\bigcup_{f\in\mathcal{F}} A_f\right) \leq \sum_{f\in\mathcal{F}} \mathbb{P}(A_f) \leq \frac{1}{2}$. The claim follows. ◀

We keep the concise notation from Section 4: for an instance $(G, \mathcal{T})$, $\mathcal{T} = ((s_1, t_1), \ldots, (s_k, t_k))$ of k-Steiner Orientation we refer to the vertices $s_i, t_i \in V(G)$ as $G\{s, i\}$ and $G\{t, i\}$.

Building the layers

Given an instance $(G, \mathcal{T}_G)$, our aim is to build a larger instance, so that if $S(G, \mathcal{T}) = k$ then the new one is also fully satisfiable, but otherwise the maximal fraction of terminal pairs being simultaneously satisfiable in the new instance is at most $\frac{1}{q}$.

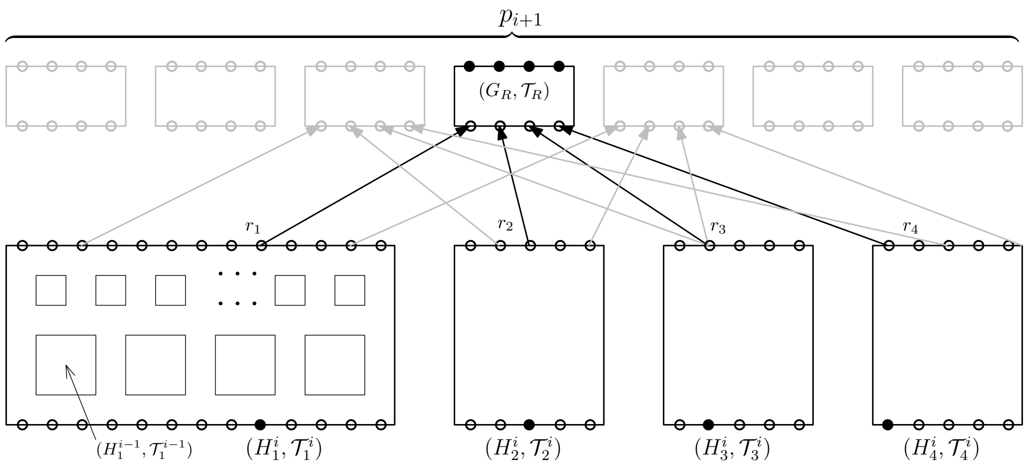

Figure 2 The construction of $(H^{i+1}, \mathcal{T}^{i+1})$ for $k = 4$. The first copy of $(H^i, \mathcal{T}^i)$ is shown in greater detail to highlight the recursive construction, with circles representing its terminal pairs. The new layer consists of p_{i+1} copies of $(G, \mathcal{T})$. The vector $R = (r_1, r_2, r_3, r_4) \in [p_i]^4$ indicates which sinks are connected to the sources in $(G_R, \mathcal{T}_R)$. The black dots show 4 terminal pairs associated with this copy. For the sake of legibility only the edges incident to G_R (the black ones) and neighboring copies (the grey ones) are sketched.

We inductively construct a family of instances $(H^i, \mathcal{T}^i)_{i=1}^B$ with $(H^1, \mathcal{T}^1) = (G, \mathcal{T}_G)$. Let $k_i = |\mathcal{T}^i|$ and p_i indicate the number of copies of $(G, \mathcal{T}_G)$ in the last layer (to be explained below) of $(H^i, \mathcal{T}^i)$. We will have that $p_1 = 1$ and $k_i = |\mathcal{T}^i| = k \cdot p_i$. We construct $(H^{i+1}, \mathcal{T}^{i+1})$ by taking k vertex-disjoint copies of the i-th instance, denoted $(H_1^i, \mathcal{T}_1^i), \ldots, (H_k^i, \mathcal{T}_k^i)$ and forming a new layer of copies of $(G, \mathcal{T}_G)$ which will be randomly connected to the i-th layer through directed edges. Therefore graph H^{i+1} will have $i + 1$ layers of copies of G.

Let $\mathcal{R} = [k_i]^k$ be the family of k-tuples $(r_1, r_2, \ldots r_k)$ with elements from the set $[k_i]$. We sample a random tuple $R = (r_1, r_2, \ldots r_k)$ from $\mathcal{R}$ and create a new copy of the original instance – let us refer to it as $(G_R, \mathcal{T}_R)$. We add a directed edge from $H_j^i\{t, r_j\}$ to $G_R\{s, j\}$ for each $j \in [k]$, thus connecting a random sink of H_j^i to the source $G_R\{s, i\}$. We insert k pairs to $\mathcal{T}^{i+1}$: $(H_1^i\{s, r_1\}, G_R\{t, 1\}), \ldots, (H_k^i\{s, r_k\}, G_R\{t, k\})$. We iterate this subroutine $p_{i+1} = \mathcal{O}(k^4 q^{2k} p_i)$ times (a derivation of this quantity is postponed to Lemma 5.4), as shown in Figure 2.

▶ **Lemma 5.4.** *Let $y_i = S(H^i, \mathcal{T}^i) \,/\, k_i$ be the maximal fraction of terminal pairs that can be simultaneously satisfied in $(H^i, \mathcal{T}^i)$. Suppose that $S(G, \mathcal{T}_G) < k$ and $y_i \ge \frac{1}{q}$. Then with probability at least $\frac{1}{2}$ it holds $y_{i+1} \le y_i - \frac{1}{2k} \cdot q^{-k}$.*

Proof. First observe that for each $(s_j, t_j) \in \mathcal{T}^i$, any (s_j, t_j)-path in $(H^i, \mathcal{T}^i)$ runs through i unique copies of $(G, \mathcal{T}_G)$. Therefore an (s_j, t_j)-pair is satisfied only if the corresponding i terminal pairs in those copies are satisfied. Recall that we have connected each copy of $(G, \mathcal{T}_G)$ in the i-th layer to the terminals from the previous layer according to a random tuple $R = (r_1, r_2, \ldots r_k) \in \mathcal{R}$.

We will now analyze how the possible orientations $\tilde{H}_1^i, \ldots, \tilde{H}_k^i$ influence the status of the terminal pairs in $\mathcal{T}^{i+1}$. Let $C_j \subseteq [k_i]$ encode which of the terminal pairs are reachable in $\tilde{H}_j^i$, that is, $C_j = \{\ell \in [k_i] \colon \tilde{H}_j^i\{t, \ell\}$ is reachable from $\tilde{H}_j^i\{s, \ell\}\}$. A tuple $C = (C_1, \ldots C_k)$ is called a *configuration* and we denote the family of all feasible configurations as $\mathcal{C}$. We have $|\mathcal{C}| \le (2^{k_i})^k = 2^{p_i k^2}$. For a configuration $C \in \mathcal{C}$ let $f_C : \mathcal{R} \to [0, 1]$ be a function describing the maximal fraction of satisfiable terminal pairs from those with sinks in G_R connected through a tuple $R \in \mathcal{R}$. Note that $f_C(R)$ depends on C, R and the best possible orientation of G_R, whereas it is oblivious to the rest of the structure of $(\tilde{H}^i_j)_{j=1}^k$. We can thus think of C as an interface between the first i layers and G_R.

For fixed orientations $\tilde{H}^i_1, \ldots, \tilde{H}^i_k$, and therefore fixed configuration $C \in \mathcal{C}$, we estimate the expected value of f_C. Let Y_j be a random Boolean variable indicating that $\tilde{H}^i_j[t, r_j]$ is reachable from $\tilde{H}^i_j[s, r_j]$ for a random $(r_1, r_2, \ldots r_k) \in \mathcal{R}$, i.e., that $r_j \in C_j$. Let $Y = \sum_{j=1}^k Y_j/k$, so that we always have $f_C(R) \le Y$.

Let $c_j = |C_j|/k_i$. By linearity of expectation $\mathbb{E}Y = \sum_{j=1}^k c_j/k$ and by the assumption $c_j \le y_i$ for all $j \in [k]$. However, with probability $\prod_{j=1}^k c_j$ we have $r_j \in C_j$ for all j, so $Y = 1$, but we cannot connect all pairs within $(G_R, \mathcal{T}_R)$ (since we assumed $S(G, \mathcal{T}_G) < k$), so $f_C(R) \le 1 - \frac{1}{k}$. This means that

$$\mathbb{E}\left(Y - f_C(R)\right) \ge \frac{1}{k} \cdot \prod_{j=1}^k c_j, \quad \text{ and so } \quad \mathbb{E}f_C(R) \le \frac{1}{k} \cdot \left(\sum_{j=1}^k c_j - \prod_{j=1}^k c_j \right).$$

The quantity $\sum_{j=1}^k c_j - \prod_{j=1}^k c_j$ can only increase when we increase some c_ℓ, because the ℓ-th partial derivative is $1 - \prod_{j \in [k],\, j \neq \ell} c_j \ge 0$. By the assumption $c_j \le y_i$ and $y_i \ge \frac{1}{q}$, hence

$$\mathbb{E}f_C(R) \le \frac{1}{k} \cdot \left(\sum_{j=1}^k y_i - \prod_{j=1}^k y_i \right) \le y_i - \frac{1}{k} \cdot q^{-k}.$$

Now we apply Lemma 5.3 for $\mathcal{F} = \{f_C : C \in \mathcal{C}\}$ and $\delta = \frac{1}{2k} \cdot q^{-k}$ to argue that for our choice of p_{i+1} – the number of copies in the last layer – the estimation works for all $C \in \mathcal{C}$ at once. The quantity $M = \mathcal{O}(\delta^{-2} \log(|\mathcal{F}|))$ in Lemma 5.3 becomes $\mathcal{O}(k^4 q^{2k} p_i)$, which is exactly as we defined p_{i+1}. We have sampled p_{i+1} tuples from $\mathcal{R}$ (let us denote this multiset as $\mathcal{R}_H \subseteq \mathcal{R}$) and added a copy $(G_R, \mathcal{T}_R)$ for each $R \in \mathcal{R}_H$. For a fixed $C \in \mathcal{C}$, the maximal fraction of satisfiable terminal pairs in $(H^{i+1}, \mathcal{T}^{i+1})$ equals the average of $f_C(R)$ over $R \in \mathcal{R}_H$. By Lemma 5.3 we know that, regardless of the choice of C, this quantity is at most

$$\mathbb{E}_{R \sim U(\mathcal{R}_H)} f_C(R) \le \mathbb{E}_{R \sim U(\mathcal{R})} f_C(R) + \frac{1}{2k} \cdot q^{-k} \le y_i - \frac{1}{2k} \cdot q^{-k},$$

with probability at least $\frac{1}{2}$ (that is, if we have chosen $\mathcal{R}_H$ correctly). Since the upper bound works for all $C \in \mathcal{C}$ simultaneously, the claim follows. ◀

Proof of Lemma 5.1. We define $(H, \mathcal{T}_H) = (H^B, \mathcal{T}^B)$ for $B = 2kq^k$. The completeness is straightforward: if $S(G, \mathcal{T}_G) = k$, then we can orient all copies of G so that $G\{t, j\}$ is always reachable from $G\{s, j\}$ and each requested path in $S(H^i, \mathcal{T}^i)$ is given as a concatenation of respective paths in B copies of G.

To see the soundness, suppose that $S(G, \mathcal{T}_G) < k$. The sequence $(y_i)_{i=0}^B$ is non-increasing and the value of y_i is being decreased by at least $\frac{i}{2k} \cdot q^{-k}$ in each iteration, as long as $y_i \ge \frac{1}{q}$, due to Lemma 5.4. Therefore after $B = 2kq^k$ iterations we are sure to have $y_B \le \frac{1}{q}$.

To estimate the size of the instance, recall that we have $k_i = p_i k$ and $p_{i+1} = \mathcal{O}(k^4 q^{2k} p_i)$. We can assume $q \ge 2$ and so $k \le q^k$. For $B = 2kq^k$, the value of p_B becomes $\left(k^4 q^{2k}\right)^{\mathcal{O}(kq^k)} = 2^{q^{\mathcal{O}(k)} \mathcal{O}(\log q)} = 2^{q^{\mathcal{O}(k)}}$. The size of $V(H)$ is at most $k_B \cdot |V(G)|$ times the number of layers, which is B. We trivially bound $B \le 2^B \le k_B$ to obtain $|V(H)| \le |V(G)| \cdot k_B^2$.

The presented construction is randomized because we randomly choose a biased sampler family in each of the B steps. If we start with a YES-instance, then we produce a YES instance regardless of these choices, and otherwise we produce a NO instance with probability at least $2^{-B} \geq \frac{1}{k_B}$. The construction can be derandomized within running time $f(k,q) \cdot |V(G)|$ as follows. In each application of Lemma 5.3 the sizes of X and $\mathcal{F}$ are $(k_i)^k$ and $2^{p_i k^2}$, respectively, and $\delta = \frac{1}{2k} \cdot q^{-k}$, which are all bounded by a function of k and q. Therefore instead of sampling a biased sampler family, we can enumerate all $\mathcal{O}(\delta^{-2} \log(|\mathcal{F}|))$-tuples of elements from X and find one giving a biased sampler family. ◀

Adjusting the parameters

The construction above implies that we can amplify the gap to any constant q by multiplying the size of an instance by a factor depending on k and q. However, when we want to rule out a superconstant approximation factor, e.g., $\alpha(k) = \log\log k$, we would like to apply the hypothetical approximation algorithm to the instance $(H, \mathcal{T}_H)$ with parameter k_0 depending on k and q, so we additionally need to adjust q so that $\alpha(k_0(k,q)) \leq q$.

▶ **Theorem 2.1.** *Consider a function $\alpha(k) = (\log k)^{\beta(k)}$, where $\beta(k) \to 0$ is computable and non-increasing. It is* W[1]*-hard to distinguish whether for a given instance of k-*Steiner Orientation*:*

1. *there exists an orientation satisfying all k terminal pairs, or*
2. *for all orientations the number of satisfied pairs is at most $\frac{1}{\alpha(k)} \cdot k$.*

Proof. We are going to reduce the exact version of k-Steiner Orientation, which is W[1]-hard, to the version with a sufficiently large gap, with Lemma 5.1. For a fixed k we can bound k_0 by a function of q: $k_0(q) \leq 2^{q^{c \cdot k}}$ for some constant c. On the other hand, $k_0(q) \to \infty$. Given an instance $(G, \mathcal{T}_G)$ we use Lemma 5.1 with q large enough, so that $\beta(k_0(q)) \cdot c \cdot k \leq 1$. The dependency $q(k)$ is also a computable function. We get $\alpha(k_0) = (\log k_0)^{\beta(k_0)} \leq q^{c \cdot k \cdot \beta(k_0)} \leq q$.

We have obtained a new instance $(H, \mathcal{T}_H)$ of k_0-Steiner Orientation of size $f(k) \cdot |V(G)|$ and k_0 being a function of k. If the original instance is fully satisfiable then the same holds for $(H, \mathcal{T}_H)$ and otherwise $S(H, \mathcal{T}_H) \leq \frac{1}{q} \cdot k_0 \leq \frac{1}{\alpha(k_0)} \cdot k_0$, which finishes the reduction. ◀

If we restrict the running time to be purely polynomial, we can slightly strengthen the lower bound, i.e., replace k with n in the approximation factor, while working with a similar hardness assumption. To make this connection, we observe that in order to show that a problem is in FPT, it suffices to solve it in polynomial time for some superconstant bound on the parameter.

▶ **Proposition 5.5.** *Consider a parameterized problem $\Pi \in$ XP that admits a polynomial-time algorithm (resp. false-biased polynomial-time algorithm) for the case $f(k) \leq |I|$, where f is some computable function. Then Π admits an* FPT *algorithm (resp. false-biased* FPT *algorithm).*

Proof. Since $\Pi \in$ XP, it admits a deterministic algorithm with running time $|I|^{g(k)}$. Whenever $f(k) \leq |I|$, we execute the polynomial-time algorithm. Otherwise we can solve it in time $f(k)^{g(k)}$. ◀

The hardness assumption below is slightly stronger than in Theorem 2.1, because the quantity 2^{k_0} can be super-polynomial and we cannot afford the time-expensive derandomization.

▶ **Theorem 2.2.** *Consider a function $\alpha(n) = (\log n)^{\beta(n)}$, where $\beta(n) \to 0$ is computable and non-increasing. Unless the class W[1] admits false-biased FPT algorithms, there is no polynomial-time algorithm that, given an instance of* Steiner Orientation *with n vertices and k terminal pairs, distinguishes between the following cases:*

1. *there exists an orientation satisfying all k terminal pairs, or*
2. *for all orientations the number of satisfied pairs is at most $\frac{1}{\alpha(n)} \cdot k$.*

Proof. Suppose there is such an algorithm with approximation factor $\alpha(n) = (\log n)^{\beta(n)}$. Let $\beta^*(\ell)$ be the smallest integer L, for which $\beta(L) \leq \frac{1}{\ell}$. The function β^* is well defined and computable, because β is computable. Again, for fixed k and some constant c we have dependency $k_0(q) \leq 2^{q^{c \cdot k}}$.

We are going to use the polynomial-time algorithm to solve k-Steiner Orientation in randomized $f(k) \cdot n^{\mathcal{O}(1)}$ time, which would imply the claim. Since the problem is in XP [15], by Fact 5.5 it suffices to solve instances satisfying $\beta^*((ck)^2) \leq n$ in polynomial time. We can thus assume $(ck)^2 \cdot \beta(n) \leq 1$, or equivalently $ck \cdot \beta(n) \leq \sqrt{\beta(n)}$.

Given an instance of k-Steiner Orientation, we apply Lemma 5.1 with $q = (2\log n)^{\beta(n)}$.

$$k_0 \leq 2^{q^{ck}} = 2^{(2\log n)^{ck \cdot \beta(n)}} \leq 2^{(2\log n)^{\sqrt{\beta(n)}}} = n^{o(1)},$$

$$\alpha(|V(H)|) \leq \alpha(n \cdot k_0^2) = \alpha(n^{1+o(1)}) = ((1 + o(1)) \cdot \log n)^{\beta(n)}.$$

For large n we obtain $\alpha(|V(H)|) \leq q$. Since $|V(H)| \leq n \cdot k_0^2 = n^{1+o(1)}$, the size of the new instance of polynomially bounded and thus the randomized construction takes polynomial time. If we started with a YES instance, we always produce a YES instance, and otherwise $S(H, \mathcal{T}_H) \leq \frac{1}{q} \cdot k_0 \leq \frac{1}{\alpha(|V(H)|)} \cdot k_0$ with probability at least $\frac{1}{k_0}$, so we need to repeat the procedure $k_0 = n^{o(1)}$ times to get a constant probability of creating an instance with a small optimum. A hypothetical algorithm distinguishing these cases would therefore entail a false-biased polynomial-time algorithm for Steiner Orientation for the case $\beta^*((ck)^2) \leq n$. The claim follows from Proposition 5.5. ◀

6 Inapproximability of Directed Multicut

We switch our attention to the Max (k,p)-Directed Multicut problem, for which we provide a slightly simpler reduction. We keep the same convention as before: within graph G we refer to sources and sinks $(s_i, t_i) \in \mathcal{T}$ shortly as $G\{s, i\}$, $G\{t, i\}$, and denote the maximal number of terminal pairs separable in $(G, \mathcal{T})$ by deleting p edges by $S(G, \mathcal{T}, p)$.

▶ **Lemma 6.1.** *There is a procedure that, for an instance $(G, \mathcal{T}_G, p)$, $|\mathcal{T}_G| = 4$ of* Directed Multicut *and parameter q, constructs a new instance $(H, \mathcal{T}_H, p_0)$, $k_0 = |\mathcal{T}_H|$, such that:*

1. *$k_0 = \Theta(p \cdot q^2 \log q)$,*
2. *$p_0 = \Theta(p^2 \log q)$,*
3. *$|E(H)| = |E(G)| \cdot p_0 + \mathcal{O}(k_0 \cdot p_0)$,*
4. *if $S(G, \mathcal{T}_G, p) = 4$, then $S(H, \mathcal{T}_H, p_0) = k_0$ always (Completeness),*
5. *if $S(G, \mathcal{T}_G, p) < 4$, then $S(H, \mathcal{T}_H, p_0) \leq \frac{1}{q} \cdot k_0$ with probability at least $\frac{1}{2}$ (Soundness).*

The randomized construction takes time proportional to $|H|$. It can be derandomized in time $f(p, q) \cdot |G|$.

Proof. Consider $M = 3(p + 1) \cdot \log q$ copies of $(G, \mathcal{T}_G)$, denoted $(G_1, \mathcal{T}_1), \ldots, (G_M, \mathcal{T}_M)$. Let $\mathcal{R} = [4]^M$ be the family of all M-tuples with values in $[4]$. For a random sequence $R = (r_1, r_2, \ldots r_M) \in \mathcal{R}$, we add a terminal pair s_R, t_R and for each $i \in [M]$ we add directed

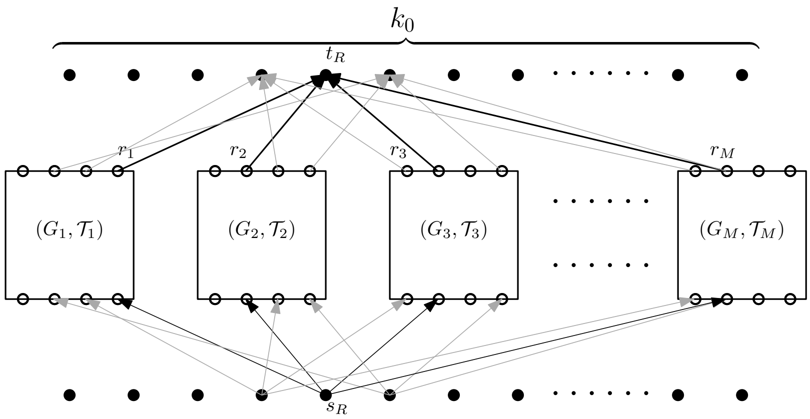

Figure 3 Building a new instance from M parallel copies of G. The tuple $R = (r_1, r_2, \dots r_M)$ encodes through which nodes the (s_R, t_R) pair is connected. For the sake of legibility only the edges incident to s_R, t_R (the black ones) and neighboring terminals (the grey ones) are sketched.

edges $(s_R, G_i\{s, r_i\})$ and $(G_i\{t, r_i\}, t_R)$. We repeat this subroutine $k_0 = \Theta(p \cdot q^2 \log q)$ times and create that many terminals pairs as depicted in Figure 3. We set the budget $p_0 = 3p(p+1) \cdot \log q$.

If $S(G, \mathcal{T}_G, p) = 4$, then the budget suffices to separate all terminal pairs in all copies of $(G, \mathcal{T}_G)$ (completeness). Otherwise, one needs to remove at least $p+1$ edges from each copy of $(G, \mathcal{T}_G)$ to separate all 4 pairs so we can afford that in at most $3p \cdot \log q$ copies. Therefore for any solution there are at least $3 \log q$ copies, where there is at least one terminal pair that is not separated.

Let $C_i \subseteq [4]$ represent information about the status of solution within $(G_i, \mathcal{T}_i)$: there is path from $G_i\{s, j\}$ to $G_i\{t, j\}$ only if $j \in C_i$. A tuple $C = (C_1, \dots, C_M)$ is called a configuration and we refer to the family of configurations induced by possible solutions as $\mathcal{C}$. Clearly, $|\mathcal{C}| \le 16^M$.

Recall that each terminal pair can be represented by a tuple $R = (r_1, r_2, \dots r_M) \in \mathcal{R}$ encoding through which terminal pair in G_i a path from s_R to t_R can go. For a fixed configuration $C \in \mathcal{C}$, function $f_C : \mathcal{R} \to \{0, 1\}$ is set to 1 if the pair s_R, t_R is separated, or equivalently: if for each $i \in [M]$ we have $r_i \notin C_i$. For $S(G, \mathcal{T}_G, p) < 4$ there are at least $3 \log q$ copies of G_i with $C_i \ne \emptyset$, therefore $\mathbb{E}_{R \sim U(\mathcal{R})} f_C(R) \le (\frac{3}{4})^{3 \log q} \le 2^{-\log(2q)} = \frac{1}{2q}$.

The size of $\mathcal{C}$ is at most $16^M = 2^{\mathcal{O}(p \log q)}$. We apply Lemma 5.3 for $\mathcal{F} = \{f_C : C \in \mathcal{C}\}$ and $\delta = \frac{1}{2q}$. It follows that $\mathcal{O}(\delta^{-2} \log(|\mathcal{F}|)) = \mathcal{O}(p \cdot q^2 \log q)$ random samples from $\mathcal{R}$ suffice to obtain a rounding error of at most $\frac{1}{2q}$ for all $C \in \mathcal{C}$ at once. Therefore with probability at least $\frac{1}{2}$ we have constructed an instance in which for any cutset of size p_0 (and thus for any configuration C) the fraction of separated terminal pairs is at most $\mathbb{E}_{R \sim U(\mathcal{R})} f_C(R) + \frac{1}{2q} \le \frac{1}{q}$. ◀

Remark on derandomization

As before, if we allow exponential running time with respect to p and q, we can find a correct sampler family by enumeration and derandomize the reduction. However, we cannot afford that in a polynomial-time reduction. To circumvent this, observe that we upper bound the expected value of f_C using independence of $3 \log q$ variables. We could alternatively take advantage of δ-biased ℓ-wise independent hashing [37] (cf. [8, 39, 40]) to construct biased ℓ-wise independent binary random variables with few random bits, instead of relying on Lemma 5.3. This technique provides an analogous bound on additive estimation error as in Lemma 5.3 for events that depend on at most ℓ variables. A family of N such variables can be constructed using $\mathcal{O}(\ell + \log\log N + \log(\frac{1}{\delta}))$ random bits [37, Lemma 4.2].

Since we are interested in having $N = \mathcal{O}(p \cdot \log q)$ variables, $\delta = \frac{1}{2q}$, and $(3\log q)$-wise independency, the size of the whole probabilistic space becomes $2^{\mathcal{O}(\log q + \log\log p)} = q^{\mathcal{O}(1)}(\log p)^{\mathcal{O}(1)}$. The problem is that we need to optimize the exponent at q in order to obtain better lower bounds. Unfortunately, we are not aware of any construction of a δ-biased ℓ-wise independent hash family, that would optimize this constant.

▶ **Theorem 2.4.** *For any $\varepsilon > 0$ and function $\alpha(k) = \mathcal{O}\left(k^{\frac{1}{2}-\varepsilon}\right)$, it is* W[1]*-hard to distinguish whether for a given instance of* Max (k,p)-Directed Multicut*:*
1. *there is a cut of size p that separates all k terminal pairs, or*
2. *all cuts of size p separate at most $\frac{1}{\alpha(k)} \cdot k$ terminal pairs.*

Proof. Let us fix $\varepsilon > 0$. We are going to reduce the exact version of p-Directed Multicut with 4 terminals, which is W[1]-hard, to the version with a sufficiently large gap, parameterized by both p and $k = |\mathcal{T}|$. Let L be an integer larger than $\frac{2}{\varepsilon}$.

For an instance $(G, \mathcal{T}, p)$ of p-Directed Multicut we apply Lemma 6.1 with $q = p^L$. If the original instance is fully solvable, the new one is as well. Otherwise the maximal fraction of separated terminal pairs is $\frac{k_0}{q} = \mathcal{O}(p \cdot q \log q) = \mathcal{O}(p^{L+2})$. On the other hand, $\frac{k_0}{\alpha(k_0)} = \Omega\left(k_0^{(\frac{1}{2}+\varepsilon)}\right) \geq \Omega\left(p^{2L\cdot(\frac{1}{2}+\varepsilon)}\right)$. The exponent at p in the latter formula is $L + 2\varepsilon L > L + 2$, so for large p it holds $\frac{k_0}{\alpha(k_0)} \geq \frac{k_0}{q}$, therefore the reduction maps NO-instances into those where all cuts of size p_0 separate at most $\frac{k_0}{\alpha(k_0)}$ terminal pairs. Both k_0 and p_0 are functions of p, therefore we have obtained a parameterized reduction. ◀

▶ **Theorem 2.5.** *Assuming* NP $\not\subseteq$ co-RP*, for any $\varepsilon > 0$ and function $\alpha(m) = \mathcal{O}\left(m^{\frac{1}{2}-\varepsilon}\right)$, there is no polynomial-time algorithm that, given an instance $(G, \mathcal{T}, p)$, $|\mathcal{T}| = k$, $|E(G)| = m$, of* Max Directed Multicut*, distinguishes between the following cases:*
1. *there is a cut of size p that separates all k terminal pairs, or*
2. *all cuts of size p separate at most $\frac{1}{\alpha(m)} \cdot k$ terminal pairs.*

Proof. Suppose there is such an algorithm for some $\varepsilon > 0$ and proceed as in the proof of Theorem 2.4 with L sufficiently large, so that $2\varepsilon L \geq 5$ and $q = m^L$. The reduction is polynomial because L is constant for fixed ε. We have $m_0 = |E(H)| = m \cdot p_0 + \mathcal{O}(k_0 \cdot p_0) = mp^2 \log q + \mathcal{O}(p^3 q^2 \log^2 q) = \mathcal{O}(m^{2L+5})$ because $p \leq m$. If the initial instance is fully satisfiable, then always $S(H, \mathcal{T}_H, p_0) = k_0$. For a NO-instance, we have $S(H, \mathcal{T}_H, p_0) \leq \frac{k_0}{q} = \mathcal{O}(m^{L+2})$ with probability at least $\frac{1}{2}$. On the other hand, $k_0 = \Omega(m^{2L})$ and

$$\frac{k_0}{\alpha(m_0)} = \Omega\left(\frac{1}{m_0^{\frac{1}{2}-\varepsilon}}\right) \cdot k_0 = \Omega\left(m^{2L-(2L+5)\cdot(\frac{1}{2}-\varepsilon)}\right) = \Omega(m^{L-\frac{5}{2}+2\varepsilon L}).$$

We have adjusted L to have $L - \frac{5}{2} + 2\varepsilon L > L + 2$, so for large m we get $\frac{k_0}{\alpha(m_0)} \geq \frac{k_0}{q}$. Therefore the reduction maps NO-instances into those where all cuts of size p_0 separate at most $\frac{k_0}{\alpha(m_0)}$ terminal pairs. When the reduction from Lemma 6.1 is correct (with probability at least $\frac{1}{2}$), we are able to detect the NO-instances. This implies that Directed Multicut $\in$ co-RP. ◀

7 Final remarks and open problems

I would like to thank Pasin Manurangsi for helpful discussions and, in particular, for suggesting the argument based on Chernoff bound in Lemma 5.1, which is surprisingly simple and powerful. A question arises whether one can derandomize this argument efficiently and construct a δ-biased sampler family in an explicit way. This would allow us to replace the assumption NP $\not\subseteq$ co-RP with P $\neq$ NP for Directed Multicut. This technique may also find use in other reductions in parameterized inapproximability.

An obvious question is if any of the studied problems admits an $o(k)$-approximation, or if the lower bounds can be strengthened. Note that for the maximization version of Directed Multicut we do not know anything better than $\frac{k}{2}$-approximation as we cannot solve the exact problem for $k > 2$. For Steiner Orientation, the reason why the value of the parameter in the self-reduction becomes so large, is that in each step we can add only an exponentially small term to the gap. Getting around this obstacle should lead to better lower bounds. Also, the approximation status for k-Steiner Orientation on planar graphs remains unclear [11]. Here we still cannot rule out a constant approximation and there are no upper bounds known.

Finally, it is an open quest to establish relations between other hard parameterized problems and their gap versions. Is $F(k)$-Gap k-Clique W[1]-hard for $F(k) = o(k)$ and is $F(k)$-Gap k-Dominating Set W[2]-hard for any function F (open questions in [26])? Or can it be possible that $F(k)$-Gap k-Dominating Set is in W[1] for some function F?

References

1 Marek Adamczyk, Jaroslaw Byrka, Jan Marcinkowski, Syed M. Meesum, and Michal Wlodarczyk. Constant-Factor FPT Approximation for Capacitated k-Median. In *27th Annual European Symposium on Algorithms (ESA 2019)*, volume 144 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ESA.2019.1.

2 Amit Agarwal, Noga Alon, and Moses S. Charikar. Improved approximation for directed cut problems. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 671–680, New York, NY, USA, 2007. ACM. doi:10.1145/1250790.1250888.

3 Esther M. Arkin and Refael Hassin. A note on orientations of mixed graphs. *Discrete Applied Mathematics*, 116(3):271–278, 2002. doi:10.1016/S0166-218X(01)00228-1.

4 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, May 1998. doi:10.1145/278298.278306.

5 Nikhil Bansal and Subhash Khot. Optimal long code test with one free bit. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 453–462. IEEE, 2009.

6 Arnab Bhattacharyya, Suprovat Ghoshal, Karthik C. S., and Pasin Manurangsi. Parameterized intractability of Even Set and Shortest Vector Problem from Gap-ETH. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pages 17:1–17:15, 2018. doi:10.4230/LIPIcs.ICALP.2018.17.

7 Nicolas Bousquet, Jean Daligault, and Stéphan Thomassé. Multicut is FPT. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 459–468, New York, NY, USA, 2011. ACM. doi:10.1145/1993636.1993698.

8 J.Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979. doi:10.1016/0022-0000(79)90044-8.

9 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 *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 743–754, October 2017. doi:10.1109/FOCS.2017.74.

10 Rajesh Chitnis, Andreas Emil Feldmann, and Pasin Manurangsi. Parameterized approximation algorithms for bidirected steiner network problems. In *26th Annual European Symposium on Algorithms (ESA 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

11 Rajesh Chitnis, Andreas Emil Feldmann, and Ondřej Suchý. A tight lower bound for Planar Steiner Orientation. *Algorithmica*, May 2019. doi:10.1007/s00453-019-00580-x.

12 Julia Chuzhoy and Sanjeev Khanna. Polynomial flow-cut gaps and hardness of directed cut problems. *J. ACM*, 56(2):6:1–6:28, April 2009. doi:10.1145/1502793.1502795.

13 Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT Approximations for k-Median and k-Means. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42:1–42:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2019.42.

14 Vincent Cohen-Addad and Jason Li. On the Fixed-Parameter Tractability of Capacitated Clustering. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41:1–41:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2019.41.

15 Marek Cygan, Guy Kortsarz, and Zeev Nutov. Steiner forest orientation problems. *SIAM Journal on Discrete Mathematics*, 27(3):1503–1513, 2013. doi:10.1137/120883931.

16 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3), June 2007. doi:10.1145/1236457.1236459.

17 Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:128, 2016.

18 Britta Dorn, Falk Hüffner, Dominikus Krüger, Rolf Niedermeier, and Johannes Uhlmann. Exploiting bounded signal flow for graph orientation based on cause–effect pairs. In *Theory and Practice of Algorithms in (Computer) Systems*, pages 104–115, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

19 Guillaume Fertin, Hafedh Mohamed-Babou, and Irena Rusu. On the Complexity of two Problems on Orientations of Mixed Graphs. In *In Proc. 5èmes Journées Ouvertes Biologie Informatique Mathématiques (JOBIM 2012)*, pages 161–170, Rennes, France, July 2012. URL: https://hal.archives-ouvertes.fr/hal-00826863.

20 Iftah Gamzu, Danny Segev, and Roded Sharan. Improved orientations of physical networks. In *Algorithms in Bioinformatics*, pages 215–225, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

21 David Gillman. A Chernoff bound for random walks on expander graphs. *SIAM J. Comput.*, 27:1203–1220, 1998.

22 Anupam Gupta, Euiwoong Lee, and Jason Li. Faster exact and approximate algorithms for k-Cut. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 113–123, 2018. doi:10.1109/FOCS.2018.00020.

23 Venkatesan Guruswami, Rajsekar Manokaran, and Prasad Raghavendra. Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 573–582. IEEE, 2008.

24 Rafael Hassin and Nimrod Megiddo. On orientations and shortest paths. *Linear Algebra and its Applications*, 114-115:589–602, 1989. doi:10.1016/0024-3795(89)90481-3.

25 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

26 Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating Dominating Set. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1283–1296, 2018. doi:10.1145/3188745.3188896.

27 Ken-ichi Kawarabayashi and Bingkai Lin. A nearly 5/3-approximation FPT algorithm for Min-k-Cut. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms*, pages 990–999, 2020. doi:10.1137/1.9781611975994.59.

28 Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 767–775, New York, NY, USA, 2002. ACM. doi:10.1145/509907.510017.

29 Bingkai Lin. The parameterized complexity of k-Biclique. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 605–615, 2015. doi:10.1137/1.9781611973730.41.

30 Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Parameterized complexity and approximability of directed odd cycle transversal. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2181–2200, 2020. doi:10.1137/1.9781611975994.134.

31 Rajsekar Manokaran, Joseph Naor, Prasad Raghavendra, and Roy Schwartz. SDP gaps and UGC hardness for multiway cut, 0-extension, and metric labeling. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 11–20, 2008.

32 Pasin Manurangsi. Almost-polynomial ratio ETH-hardness of approximating Densest k-Subgraph. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 954–961, New York, NY, USA, 2017. ACM. doi:10.1145/3055399.3055412.

33 Pasin Manurangsi. Inapproximability of Maximum Edge Biclique, Maximum Balanced Biclique and Minimum k-Cut from the Small Set Expansion Hypothesis. In *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 79:1–79:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2017.79.

34 Pasin Manurangsi and Prasad Raghavendra. A Birthday Repetition Theorem and Complexity of Approximating Dense CSPs. In *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 78:1–78:15, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2017.78.

35 Daniel Marx. Completely inapproximable monotone and antimonotone parameterized problems. In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 181–187, June 2010. doi:10.1109/CCC.2010.25.

36 Alexander Medvedovsky, Vineet Bafna, Uri Zwick, and Roded Sharan. An algorithm for orienting graphs based on cause-effect pairs and its applications to orienting protein networks. In *Algorithms in Bioinformatics*, pages 222–232, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

37 Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22, March 2001. doi:10.1145/100216.100244.

38 Marcin Pilipczuk and Magnus Wahlström. Directed Multicut is W[1]-hard, even for four terminal pairs. *ACM Trans. Comput. Theory*, 10(3):13:1–13:18, May 2018. doi:10.1145/3201775.

39 Alan. Siegel. On universal classes of extremely random constant-time hash functions. *SIAM Journal on Computing*, 33(3):505–543, 2004. doi:10.1137/S0097539701386216.

40 Umesh Virkumar Vazirani. *Randomness, Adversaries and Computation (Random Polynomial Time)*. PhD thesis, University of California, Berkeley, 1986.