Lower Bounds for the Parameterized Complexity of Minimum Fill-in and Other Completion Problems

IVAN BLIZNETS, National Research University Higher School of Economics, Russia
MAREK CYGAN, PAWEL KOMOSA, and MICHAŁ PILIPCZUK, University of Warsaw, Poland
LUKÁŠ MACH, University of Warwick, England

In this work, we focus on several completion problems for subclasses of chordal graphs: Minimum Fill-In, Interval Completion, Proper Interval Completion, Trivially Perfect Completion, and Threshold Completion. In these problems, the task is to add at most $k$ edges to a given graph to obtain a chordal, interval, proper interval, threshold, or trivially perfect graph, respectively. We prove the following lower bounds for all these problems, as well as for the related Chain Completion problem:

- Assuming the Exponential Time Hypothesis, none of these problems can be solved in time $2^{O(n^{1/2}/\log c \cdot n)}$ or $2^{O(k^{1/4}/\log c \cdot k)} \cdot n^{O(1)}$, for some integer $c$.
- Assuming the non-existence of a subexponential-time approximation scheme for Min Bisection on $d$-regular graphs, for some constant $d$, none of these problems can be solved in time $2^{o(n)}$ or $2^{o(\sqrt{k})} \cdot n^{O(1)}$.

For all the aforementioned completion problems, apart from Proper Interval Completion, FPT algorithms with running time of the form $2^{O(\sqrt{k} \log k)} \cdot n^{O(1)}$ are known. Thus, the second result proves that a significant improvement of any of these algorithms would lead to a surprising breakthrough in the design of approximation algorithms for Min Bisection.

To prove our results, we use a reduction methodology based on combining the classic approach of starting with a sparse instance of 3-Sat, prepared using the Sparsification Lemma, with the existence of almost linear-size Probabilistically Checkable Proofs. Apart from our main results, we also obtain lower bounds excluding the existence of subexponential algorithms for the Optimum Linear Arrangement problem, as well as improved, yet still not tight, lower bounds for Feedback Arc Set in Tournaments.

CCS Concepts: • Mathematics of computing → Graph algorithms; • Theory of computation → Approximation algorithms analysis; Fixed parameter tractability;

I. Bliznets’s work was supported by the Government of the Russian Federation (Grant No. 14.Z50.31.0030), as well as by Warsaw Center of Mathematics and Computer Science. M. Cygan’s work was partially supported by the Polish National Science Center Grant No. DEC-2012/05/D/ST6/03214. L. Mach’s work was funded by the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007-2013)/ERC Grant No. 259385. M. Pilipczuk’s work was supported by the Polish National Science Center Grant No. DEC-2013/11/D/ST6/03073. During the work on these results, M. Pilipczuk was holding a post-doc position of Warsaw Center of Mathematics and Computer Science and was supported by the Foundation for Polish Science via the START stipend programme. Moreover, the work of M. Cygan and P. Komosa was supported in part by the European Research Council’s TOTAL Grant No. 677651.

Authors’ addresses: I. Bliznets, National Research University Higher School of Economics, 16 Soyuza Pechatnikov Street, 190121, St. Petersburg, Russia; email: iabliznets@gmail.com; M. Cygan, P. Komosa, and M. Pilipczuk, University of Warsaw, Krakowskie Przedmieście 26/28, 00-927, Warsaw, Poland; email: cygan@mimuw.edu.pl, kompaw01@gmail.com, mp248287@mimuw.edu.pl; L. Mach, University of Warwick, CV4 7AL, Coventry, England; email: lukas.mach@gmail.com. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.
1549-6325/2020/03-ART25 $15.00
https://doi.org/10.1145/3381426

ACM Transactions on Algorithms, Vol. 16, No. 2, Article 25. Publication date: March 2020.
1 INTRODUCTION

In the MINIMUM FILL-IN problem, also known as CHORDAL COMPLETION, the input is an undirected graph $G$ and an integer $k$, and the question is whether at most $k$ edges can be added to $G$ to turn it into a chordal graph, i.e., a graph without induced cycles of length at least 4 (also known as holes). The interest in this problem originates in the study of strategies for Gaussian elimination on sparse matrices, because the optimum number of additional entries of a matrix $A$ that become non-zero during the elimination is tightly connected to the minimum fill-in of the graph $G_A$ obtained by taking $A$ to be its adjacency matrix. See References [16, 41] for more information on applications of MINIMUM FILL-IN in the theory of sparse matrices. However, the problems of adding as few edges as possible to obtain a chordal graph, or a graph belonging to some natural subclass of chordal graphs, like interval, proper interval, trivially perfect, or threshold graphs, have numerous other applications ranging from database management, bioinformatics, artificial intelligence, to social networks. We refer to the introductory sections of References [6, 7, 19, 21, 25, 32, 43] for a broader discussion and pointers to relevant literature.

MINIMUM FILL-IN is NP-hard, as shown by Yannakakis [44]; however, the reduction showing this is quite non-obvious; the complexity status of the problem was among the 12 open problems at the end of the first edition of the Garey and Johnson’s book [26]. The study of MINIMUM FILL-IN from the point of view of parameterized complexity, with $k$ being the obvious parameter of interest, started with the pioneering work of Kaplan et al. [32]. They proposed a fixed-parameter (FPT) algorithm with running time $O(16^k k^6 + k^2 mn)$ that is based on locating holes in the graph and branching on possible ways of adding edges to get rid of them. A similar strategy worked also for PROPER INTERVAL COMPLETION [32], the problem of adding as few edges as possible to obtain a proper interval graph, but not for INTERVAL COMPLETION. The fixed-parameter tractability of the latter has been resolved by Villanger et al. [43] only several years thereafter.

A complete turning point came from Fomin and Villanger [25], who presented an algorithm for MINIMUM FILL-IN with subexponential parameterized complexity, more precisely with running time $O(2^{O(\sqrt{k \log k})} + k^2 nm)$. This was an immense surprise to the parameterized complexity community, since subexponential parameterized algorithms, i.e., with running time $2^{o(k)} \cdot n^{O(1)}$, were known essentially only in two restricted settings: in topologically constrained graphs via the technique of bidimensionality (see, e.g., Reference [17]), and in tournaments, with an important example of FEEDBACK ARC SET IN TOURNAMENTS [3, 23, 33]. For most natural parameterized problems, the existence of such algorithms can be excluded under the Exponential Time Hypothesis, which (when combined with the Sparsification Lemma [31]) essentially states that there is no algorithm for 3-SAT with running time $2^{o(n+m)}$; cf. References [15, 24].

The work of Fomin and Villanger [25] presented a conceptual breakthrough in the approach to completion problems for subclasses of chordal graphs. The main idea is not to focus on breaking individual obstacles such as holes by single edge additions, as was proposed in the previous works, but to build a structural decomposition of the completed graph by means of a dynamic programming algorithm that minimizes the number of edges missing in the decomposition. The crux is to show that this dynamic programming can be restricted to a space of states that has size...
2^{O(\sqrt{E \log k})} \cdot n^{O(1)}, and can be enumerated efficiently. In case of Minimum Fill-In, the considered decomposition is the clique tree, but this generic approach can be in principle applied to every subclass of chordal graphs whose graphs have a well-defined global structure. And so, following [25], subexponential parameterized algorithms have been designed for Threshold Completion [20], Trivially Perfect Completion [20], Proper Interval Completion [6], and even Interval Completion [7]. Apart from Proper Interval Completion, for all these problems the algorithms have running time 2^{O(\sqrt{E \log k})} \cdot n^{O(1)}; for the former, we are currently stuck at exponent \(O(k^{2/3} \log k)\), but this is conjectured to be an artifact of the technique [6]. Let us remark that in each of these cases the idea of Fomin and Villanger only provides the basic outline of the strategy, while the actual implementation is always class-specific and requires involved technical insight.

Drange et al. [20] complemented their results with a number of lower bounds suggesting that even a slight deviation from the setting of adding edges to a subclass of chordal graphs leads to the non-existence of a subexponential parameterized algorithm under the assumption of ETH. Thus, the aforementioned problems for which such algorithms exist are in fact intriguing “singularity points” on the complexity landscape. It is interesting that these singularity points actually correspond to problems that have important practical applications. Hence, the study of this phenomenon is an important direction that naturally belongs to the optimality programme (cf. References [35, 38]): a trend in parameterized complexity that focuses on systematic study of parameterized problems by providing possibly tight upper and lower bounds on their complexity.

From this point of view, the first natural question is exactly how “deep” are the aforementioned singularity points. More precisely, is the running time of the form 2^{O(\sqrt{E \log k})} \cdot n^{O(1)} optimal, say under ETH? For subexponential algorithms derived using bidimensionality, the non-existence of 2^{o(\sqrt{E})} \cdot n^{O(1)}-time algorithms under ETH usually follows already from the known NP-hardness reductions [15, 24]. However, for the considered completion problems this is not the case: as Fomin and Villanger already observed in Reference [25], the classic NP-hardness reduction for Minimum Fill-In gives only a 2^{o(k^{1/6})} \cdot n^{O(1)} lower bound. A similar situation holds for Feedback Arc Set in Tournaments, for which the fastest known algorithms work in time 2^{O(\sqrt{E})} \cdot n^{O(1)} [23, 33].

For this reason, the question of establishing tight upper and lower bounds for Minimum Fill-In was already asked explicitly by Fomin and Villanger [25], repeated by Marx in his survey on the optimality programme [38], and then reiterated for respective subclasses in all the works [6, 7, 20]. The goal of this article is to remedy this situation by providing complexity foundations for proving that the square root in the exponent of the running time is hard to improve.

Our results. First, we investigate how strong lower bounds for completion problems can be obtained when we assume only ETH.

\textbf{Theorem 1.1.} Unless ETH fails, there is an integer c ≥ 1 such that there are no 2^{O(\sqrt{E \log k})} \cdot n^{O(1)} algorithms for the following problems: Minimum Fill-In, Interval Completion, Proper Interval Completion, Trivially Perfect Completion, Threshold Completion, Chain Completion.

Here, a graph \(G\) is a \textit{chain graph} if it is a bipartite graph with a fixed bipartition \(A \sqcup B\), where the vertices of \(A\) can be ordered by a linear ordering \(\preceq\) such that \(N(u) \subseteq N(v)\) whenever \(u \preceq v\), for all \(u, v \in A\). In the Chain Completion problem, we are given a bipartite graph with a fixed bipartition \(A \sqcup B\), and we want to add at most \(k\) edges between \(A\) and \(B\) to obtain a chain graph. Chain Completion also admits an algorithm with running time \(O(2^{O(\sqrt{E \log k})} + k^2 nm)\) [25].

Unfortunately, Theorem 1.1 does not provide a tight result for completion problems we are interested in: We still have a gap between \(k^{1/4}\) and \(k^{1/2}\). We in fact prove a higher lower bound,
but we need to ground it on a stronger complexity assumption. More precisely, we consider the approximability of the Min Bisection problem on \( d \)-regular graphs: given a \( d \)-regular graph \( G \) with an even number of vertices, find a partition of \( V(G) \) into equal-sized parts that minimizes the number of edges between the parts. For \( 0 \leq \alpha < \beta \leq 1 \), problem Gap Min Bisection(\( d \)|\( \alpha, \beta \)) is defined as the gap problem of deciding whether the optimum solution is at most \( \alpha m \), or at least \( \beta m \).

**Hypothesis 1.2.** There exist \( 0 \leq \alpha < \beta \leq 1 \), and an integer \( d > \frac{4}{\beta - \alpha} \), such that there is no \( 2^{o(n)} \)-time algorithm for Gap Min Bisection(\( d \)|\( \alpha, \beta \)).

**Theorem 1.3.** Unless Hypothesis 1.2 fails, there is no \( 2^{o(n+m)} \)-time algorithm for Chain Completion, and no \( 2^{o(n)} \)-time algorithms for Minimum Fill-In, Interval Completion, Proper Interval Completion, Trivially Perfect Completion, and Threshold Completion. Consequently, none of these problems can be solved in time \( 2^{o(\sqrt{k})} \cdot n^{O(1)} \).

Thus, Theorem 1.3 asserts that the existence of a substantially faster algorithm for any of the considered completion problems would lead to a breakthrough in the design of approximation algorithms for Min Bisection. We do not intend to take a stance on whether Hypothesis 1.2 is true or false. However, the existence of an algorithm refuted by Theorem 1.2 is far from the current knowledge: the best polynomial-time approximation algorithms for Min Bisection achieve approximation factor \( O(\log OPT) \) [40], whereas to refute Hypothesis 1.2 one would need to obtain approximation factor arbitrarily close to one. This could be, however, possible due to the assumption of \( d \)-regularity, for a constant \( d \), as well as the access to subexponential time computations. We are not aware of any approximation algorithms for Min Bisection that would significantly use any of these assumptions. In fact, Berman and Karpinski [5] have shown that approximating Min Bisection in graphs of maximum degree 3 is as hard as in general graphs from the point of view of polynomial time approximation.

Finally, using the methodology of Theorem 1.1, we can give an improved lower bound for Feedback Arc Set in Tournaments. For this problem, the first subexponential FPT algorithm with running time \( 2^{O(\sqrt{\mathit{nf}} \log k)} \cdot n^{O(1)} \) was given by Alon et al. [3], which was later improved to \( 2^{O(\sqrt{\mathit{nf}})} \cdot n^{O(1)} \) independently by Feige [23] and by Karpinski and Schudy [33].

**Theorem 1.4.** Unless ETH fails, there is an integer \( c \geq 1 \) such that there is no \( 2^{O((\sqrt{n} / \log^c n))} \)-time algorithm for Feedback Arc Set in Tournaments.

A direct inspection of the classic NP-hardness proofs [1, 2, 14] gives only a \( 2^{O(n^{1/4})} \)-time lower bound, but Theorem 1.4 is still not tight: We have a gap between \( k^{1/2} \) and \( k^{1/4} \), similar as in Theorem 1.1. Unfortunately, despite efforts we are unable to close this gap using Hypothesis 1.2, or a similar assumption.

**Our techniques.** The main idea of this work is to carefully combine the standard approach of starting with a sparse instance of 3-SAT, prepared using the Sparsification Lemma [31], with the existence of almost linear Probabilistically Checkable Proofs (PCPs). More precisely, an application of the Sparsification Lemma reduces solving an instance \( \varphi \) of 3-SAT to solving a number of instances \( \varphi_1, \varphi_2, \ldots, \varphi_{t} \) of 3-SAT on the same variable sets, but in each \( \varphi_i \), we have that the number of clauses \( m \) is linear in the number of variables \( n \). Then, to each \( \varphi_i \), we apply the result of Dinur [18] on the existence of almost linear-size PCPs. This transforms each \( \varphi_i \) into an instance \( \varphi_i' \) of 3-SAT where

\[\]
the number of variables and clauses is \( O(n \log^c n) \), but the instance is either satisfiable, or at most a \( \rho \)-fraction of clauses can be satisfied, for some \( \rho < 1 \).

Having such an instance at hand, we can proceed with the chain of reductions proposed by Garey et al. [27] and Yannakakis [44]. Namely, we first use simple manipulations to obtain an equivalent instance of \textsc{Max Cut} with a gap, and then reduce it to the \textsc{Optimum Linear Arrangement (OLA)} problem: Given a graph \( G \) on \( n \) vertices, find a linear ordering \( \pi : V(G) \to \{1, 2, \ldots, n\} \) that minimizes the cost defined as \( \sum_{u,v \in E(G)} |\pi(u) - \pi(v)| \); value \( |\pi(u) - \pi(v)| \) is also called the cost of \( uv \). This is precisely the moment where we exploit that we are working with an instance of \textsc{Max Cut} with a gap. Namely, the construction introduces a huge clique to the complement of the \textsc{Max Cut} instance; this clique is supposed to separate in the ordering the sides of an optimum max-cut solution. In Garey et al. [27], its size must be large enough so that the cost of any edge jumping over the clique dwarfs the “noise” contribution that is given by internal ordering of parts on the left and on the right. Hence, the clique is chosen to be of size \( \Theta(n^4) \), which explodes the size of the instance. However, we observe that by starting with an instance with a gap, we can accommodate the noise in the gap, and therefore we only need the clique to be of linear size. Thus, we obtain the following theorem that can be of independent interest.

**Theorem 1.5.** Unless ETH fails, for some \( c > 1 \) there is no algorithm solving \textsc{Optimum Linear Arrangement} in time \( 2^{O(n/\log^c n)} \).

Note that OLA can be solved by a simple Held-Karp dynamic programming on subsets in time \( 2^n \cdot n^{O(1)} \), so Theorem 1.5 gives an almost tight result.

To obtain Theorem 1.1 one just has to apply a slightly modified elegant reduction from (the decision variant of) OLA to \textsc{Minimum Fill-In}, proposed by Yannakakis [44]. The reduction firstly reduces OLA to \textsc{Chain Completion}. Because of this fact, later we get a very specific \textsc{Minimum Fill-In} output instance. And this immediately gives us a reduction to \textsc{Proper Interval Completion} and \textsc{Interval Completion} problems. With slight changes, we can obtain a similar result for \textsc{Threshold Completion} and \textsc{Trivially Perfect Completion}. Theorem 1.4 is derived by applying the same methodology to the randomized hardness reduction for the problem \textsc{Feedback Arc Set in Tournaments} of Ailon et al. [1].

The moment when we lose tightness in our chain of reductions is precisely the last reduction from OLA to \textsc{Minimum Fill-In}. This is because in the construction we introduce a new vertex for every edge of the original graph, which blows up the vertex set quadratically. Therefore, the reduction would give a tight lower bound if we started with a sparse instance of \textsc{Optimum Linear Arrangement}. This is precisely our idea behind the proof of Theorem 1.3.

Namely, we observe that by starting from the hardness of \textsc{Gap Min Bisection}(\( d \)\([\alpha, \beta]\)), instead of \textsc{Max Cut}, we can give an alternative hardness reduction to \textsc{Optimum Linear Arrangement} in graphs whose degree is bounded by a constant \( d \) (\( \textsc{OLA}_\leq (d) \) for short). This reduction is much more intricate. Essentially, we replace the usage of a huge clique (which would blow up the degree) by a careful construction using several layers of expander graphs that mimics the same behaviour. This proves the following result.

**Theorem 1.6.** Unless Hypothesis 1.2 fails, there exists an integer \( d \in \mathbb{N} \) such that there is no \( 2^{o(n)} \)-time algorithm solving the decision version of \( \textsc{OLA}_\leq (d) \) in multigraphs.

From this, Theorem 1.3 follows similarly as Theorem 1.1 followed from Theorem 1.5.

**Related work.** The combination of ETH, Sparsification Lemma, and PCP tools was already used in the recent line of work on developing the theory of hardness of FPT approximation; see, e.g., References [8–11, 28, 34]. There, the main goal is to provide lower bounds on how the access
to FPT-time computations can help in the design of approximation algorithms for fundamental inapproximable problems, like \textsc{clique} or \textsc{set cover}.

The existence of almost linear PCPs combined with ETH was also used by Marx [37] to give almost tight lower bounds for the running times of polynomial time approximation schemes (PTASes) for several fundamental problems on planar graphs and in the Euclidean plane. The basic principle behind the approach of Marx [37] is very similar to ours. Namely, it is observed that using ETH and PCP tools one can prepare a sparse instance of the maximization variant of 3-Sat where it is hard to distinguish between a fully satisfiable instance and an instance where only a \((1 - \epsilon)\)-fraction of the clauses can be satisfied in time \(2^{O(n/\log^c n)}\), for some constant \(c\) (see also Theorem 3.1 in this work). This observation is used as a base for further reductions refuting the existence of certain approximation algorithms for geometric and planar problems.

Thus, this work provides another example where the said combination appears to be useful. This time, we use it to prove improved lower bounds on the complexity of FPT algorithms solving certain graph modification problems exactly. The new idea in this work is that the gap property of the considered instances can be used not only to exclude the existence of approximation algorithms but also to limit the instance size explosion in a chain of NP-hardness reductions by using more thrifty constructions.

Hypothesis 1.2 can possibly have links with the hypothesis put forward by Feige [22]. Essentially, Feige conjectures the hardness of distinguishing a “typical” 3-Sat instance from a satisfiable one in polynomial time. From this, he derives as a corollary a variant of Hypothesis 1.2, but without the assumption of \(d\)-regularity, for a constant \(d\), and with subexponential time replaced by polynomial. It is conceivable that Hypothesis 1.2 can be also implied by some stronger variant of Feige’s conjecture, but we refrain from formalizing this link due to many technical problems that arise when attempting to do this.

\textbf{Outline of the article.} In Section 2, we give all the needed definitions and list the tools used in our reductions, such as the Exponential Time Hypothesis and the PCP theorem. In Section 3, we prove Theorem 1.5. The main technical contribution of the article, that is the proof of Theorem 1.6, is contained in Section 4. Theorem 1.6 is the main ingredient in the proof of Theorem 1.3, which is presented in Section 5. Section 6 is devoted to the proof of Theorem 1.4.

2 \textbf{PRELIMINARIES}

2.1 Parameterized Complexity

A \textit{parameterized problem} \(Q\) is a subset of \(\Sigma^* \times \mathbb{N}\), for a fixed finite alphabet \(\Sigma\). An instance of the problem \(Q\) is an element \((x, k) \in \Sigma^* \times \mathbb{N}\) with the integer \(k\) called the parameter. A parameterized problem is \textit{fixed parameter tractable} if there exists an algorithm deciding whether \((x, k) \in Q\) and working in time \(f(k) \cdot \text{poly}(|x|)\) for every instance \((x, k)\), where \(f\) is a computable function. For denoting the time of an algorithm, we will use the \(O^*\) notation that suppresses factors polynomial in the size of the input, e.g., \(O^*(f(k))\).

We denote a polynomial-time deterministic linear reduction from a problem \(X\) to \(Y\) by \(X \leq_{\text{lin}}^p Y\), i.e., \(X \leq_{\text{lin}}^p Y\) means that there is a polynomial time deterministic algorithm, which given an instance \(I\) of the problem \(X\) produces an instance \(I'\) of the problem \(Y\) of size \(O(|I|)\) (where \(|I|\) is the number of bits of \(I\)), such that \(I\) is a yes-instance if and only if \(I'\) is a yes-instance.

2.2 Graph Notations

A standard graph theoretical notation is used throughout the article. A \textit{graph} \(G\) is a tuple \((V(G), E(G))\), where \(V(G)\) is the set of \textit{vertices} and \(E(G) \subseteq \binom{V(G)}{2}\) is the set of \textit{edges}. When it is clear from the context what graph we refer to, we will denote by \(V\) and \(E\) the set of vertices.
and edges of $G$. A graph $H$ is a subgraph of $G$ if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A graph $H$ is an induced subgraph of $G$ if $V(H) \subseteq V(G)$ and $E(H) = E(G) \cap \left( V(H)^2 \right)$. An induced subgraph of $G$ with vertex set $X$ is denoted by $G[X]$. The complement of $G$ is the graph with the vertices $V$ and the edges $\left( V \right) - E$, and we denote it by $\overline{G}$. If $X, Y \subseteq V$ are disjoint, then $E_G(X, Y)$ is the set of edges between $X$ and $Y$. For $X \subseteq V$, we denote by $\delta_G(X)$ the set of edges with exactly one endpoint in $X$. We define a cut, as the set of edges $E_G(A, B)$, for a partition $(A, B)$ of $V$. We denote the size of the cut by $|E_G(A, B)|$.

We often use $n$, and $m$ to denote the size of $V$ and $E$, respectively. For a vertex $v$, $\deg_G(v)$ denotes the degree (the number of incident edges) of the vertex $v$. We say that a graph is \emph{d-regular} if the degree of each vertex is equal to $d$. We denote by $\Delta_G$ the maximum degree of $G$. The set $N_G(v) = \{w : (v, w) \in E(G)\}$ is the neighbourhood of $v$. We extend this notation to subsets of vertices $X$, i.e., $N_G(X) = \bigcup_{v \in X} N_G(v) \setminus X$. We will also omit subscripts, i.e., $\deg(v)$, $\Delta$, $N(X)$, $\delta(X)$, $E(U, V)$, when it is clear from the context, which graph we refer to.

We use $G[X, Y]$ to denote the induced bipartite subgraph of $G$ with parts $X$ and $Y$. That is, $G[X, Y]$ is the graph with vertex set $X \cup Y$ that contains precisely the edges $E_G(X, Y)$.

### 2.3 Expanders

The Cheeger number $h(G)$ of a graph $G$ is defined as

$$h(G) := \min \left\{ \frac{\left| \delta(X) \right|}{|X|} : X \subseteq V(G), |X| \leq \frac{|V(G)|}{2} \right\}.$$  

We say that a graph $G$ is a $(d, e)$-expander if it is $d$-regular and has Cheeger number at least $e$. When discussing expanders, it is convenient to allow parallel edges and self-loops, which naturally appear in most of the expanders constructions. It is important to note that a self-loop contributes 1 to the degree of a vertex.

There are many efficient constructions of expanders available in the literature. The following theorem produces graphs with provably optimal Cheeger numbers:

**Theorem 2.1** ([36, 39]). Let $d = p^k + 1$, where $p$ is a prime and $k \in \mathbb{N}$, and let $q$ be a prime congruent to 1 mod 4. Assume $p \neq q$. There exists a $(d, \frac{d}{2} - \sqrt{d} - 1)$-expander on $q + 1$ vertices. Furthermore, such an expander can be constructed in polynomial time.

Throughout this article, we use expanders of various sizes; hence, the following theorem appears to be useful in our setting, despite providing a weaker bound on the Cheeger number.

**Theorem 2.2.** ([Theorem 21.19 of Reference [4]]) Let $p > 0$ be a real. Then there exists a positive integer $d$, such that for every positive integer $n$ there exists a $d$-regular multigraph $G_{n, d}$ on $n$ vertices with $h(G_{n, d}) \geq p$. Moreover, graph $G_{n, d}$ can be constructed in time polynomial in $n$.

### 2.4 Linear Arrangements

A linear arrangement of an $n$-vertex graph $G = (V, E)$ is a function (a vertex ordering) $\pi : V \to \{1, \ldots, n\}$. The cost of a linear arrangement $\pi$ is defined by $\sum_{uv \in E} |\pi(u) - \pi(v)|$. We call $\pi$ the optimum linear arrangement if its cost is minimized over all vertex orderings of $G$, and we denote this cost by OLA($G$).

### 2.5 Satisfiability

We employ a standard notation related to SAT problems. We use symbols $x_1, \ldots, x_n$ for the variables of an instance, and $C_1, \ldots, C_m$ for the clauses. An $l$-CNF formula is El-CNF, if it has exactly $l$ distinct literals. We say that an assignment of $x_1, \ldots, x_n$ NAE-satisfies an $l$-CNF formula if every clause contains a satisfied and an unsatisfied literal.
2.6 Gap Problems and PCPs

In a gap version of a problem, the input instance is promised to belong to one of two languages specifying the allowed input, and the goal is to decide which case (language) a given instance belongs to. Gap problems are associated with two parameters $\alpha, \beta$. For example, in \textbf{GapMaxCut [$\alpha, \beta$]}, we are to distinguish between the case when a given graph admits a cut of size at least $\beta m$ and the case when a given graph does not admit a cut of size larger than $\alpha m$. Similarly for satisfiability problems with gap $[\alpha, \beta]$ we are to distinguish between a formula, for which at least $\beta m$ clauses can be satisfied, and the case where it is impossible to satisfy more than $\alpha m$ clauses.

To introduce gaps in our reductions, we will use the following fundamental result of Dinur.

**Theorem 2.3 (Almost Linear Size PCP, Theorem 8.3 of Reference [18]).** $3$-Sat $\in$ \textbf{PCP$_{1, \frac{1}{2}}$}$(\log(n) + O(\log \log(n)), O(1))$, where $n$ is the number of variables in an instance of $3$-Sat.

We would like to note that in Reference [18] a verifier belonging to \textbf{PCP($r, q$)} is allowed to use $O(r)$ random bits, while for us it is of utter importance to keep track of all the multiplicative constants. An inspection of the proof of Theorem 8.3 of Reference [18] shows that when Lemma 8.5 of Reference [18] is invoked, a constraint graph is created with at most $n \log O(1) n$ vertices and edges for a constant size alphabet. A single random edge of this graph can be selected by using $\log n + O(\log \log n)$ bits, which is exactly what we need.

2.7 Exponential Time Hypothesis

The Exponential Time Hypothesis (ETH), introduced by Impagliazzo, Paturi and Zane [30, 31] is now an established tool used for proving conditional lower bounds in the parameterized complexity area (see Reference [35] for a survey on ETH-based lower bounds). Intuitively, ETH states that $3$-Sat cannot be solved in time subexponential in the number of variables.

**Hypothesis 2.4 (ETH [30, 31]).** There exists $\delta > 0$ such that there is no deterministic algorithm for $3$-Sat running in time $2^{\delta n}$.

We can also formulate ETH in a stronger form, where also randomized algorithms are ruled out. An algorithm is said to have a constant two sided error if for some constant $c > 0.5$ the algorithm with probability at least $c$ declares a yes-instance to be a yes-instance, and with probability at least $c$ declares a no-instance to be a no-instance.

**Hypothesis 2.5 (Randomized ETH [15]).** There exists $\delta > 0$, such that there is no randomized algorithm with constant two sided error for $3$-Sat running in time $2^{\delta n}$.

**Lemma 2.6 (Sparsification Lemma [31]).** For every $\epsilon > 0$, there is an algorithm that takes a 3-CNF formula $\phi$ on $n$ variables and returns $l$ 3-CNF formulas $\phi_1, \ldots, \phi_l$, such that: i) $l = O(2^{\epsilon n})$, ii) for every $i$, $\phi_i$ has $n$ variables, and every such variable appears in at most $c_\epsilon$ clauses of $\phi_i$, for some constant $c_\epsilon \geq 0$, iii) $\phi$ is satisfiable if and only if at least one $\phi_i$ is satisfiable. The running time of the algorithm is $O^*(2^{\epsilon n})$.

Consequently, based on ETH, we have subexponential hardness of $3$-Sat in terms of both the number of variables and clauses.

**Theorem 2.7 ([31]).** Unless ETH fails, there exists $\delta > 0$ such that there is no deterministic algorithm for $3$-Sat running in time $2^{\delta(n+m)}$. 
3 COMBINING KNOWN REDUCTIONS FOR OLA WITH PCPS AND ETH

In this section, we discuss the proof of Theorem 1.5, i.e., explain how to show that unless ETH fails, there is no $2^{O\left(\frac{m}{\log^3(m)}\right)}$-time algorithm for OPTIMUM LINEAR ARRANGEMENT, for some constant $c \in \mathbb{N}$. We combine several well known concepts from complexity theory: ETH, the PCP theorem, the Sparsification Lemma, gap reductions, and gap amplification.

First, we show that the combination of ETH, together with the almost linear size PCP theorem, and the Sparsification Lemma, gives subexponential hardness of Gap E3-SAT. This fact was already observed and used by Marx [37].

**Theorem 3.1.** Unless ETH fails, there exist $c \in \mathbb{N}$, and $r \in (0, 1)$ such that there is no $2^{O\left(\frac{m}{\log^3(m)}\right)}$-time algorithm for Gap E3-SAT.$_{[r, 1]}$.

**Proof.** From Theorem 2.3, it follows that there is a PCP$_{1, \frac{1}{2}}(\log(n) + A\log\log(n), B)$ non-adaptive verifier $V$ for 3-SAT, for some constants $A, B > 0$.

Assume we are given the input E3-CNF formula $\phi$. Fix any $\epsilon > 0$ and apply Lemma 2.6 to $\phi$, obtaining $I = O(2^{\epsilon n})$ instances $\phi_i, \ i = 1, \ldots, I$, each with size bounded by $cn$, for some constant $c > 0$, depending on $\epsilon$, where by size we mean the sum of sizes of all the clauses in a formula. Let $R = \log(cn) + A\log\log(cn)$.

If we take $\phi_i$ as the input to the verifier $V$, then for each random binary string of length at most $R$, it reads at most $B$ bits of the proof (we will assume without loss of generality that $V$ always reads exactly $B$ bits). Thus, $V$ has access to $N$ bits of the proof in total, where $N \leq B \cdot 2^R = B \cdot cn \log^A(cn)$. We create variables $x_1, \ldots, x_N$ corresponding to each of these bits. For a boolean string $r$ of length $R$, we define a boolean function $f_r : \{0, 1\}^B \to \{0, 1\}$, with its arguments corresponding to relevant variables $x_i, \ldots, x_{i_B}$ that can be read from the proof. $f_r$ is defined as follows, for every $S \in \{0, 1\}^B$ it evaluates to true on $S$ if and only if $V$ accepts $\phi_i$ using $S$ as the bits from the proof. Every $f_r$ can be written as an equivalent E3-CNF formula $F_r$ (possibly adding some constant number of variables). Let $C$ be an upper bound (depending on $B$) for the number of clauses for each such formula. Let $\phi'_i$ be a formula combined by AND of all $F_r$. It has the number of clauses bounded by $C \cdot 2^R$ and the following holds.

If $\phi_i$ is satisfiable, then there is a proof for which $V$ always accepts. By taking the corresponding values from this proof and setting them to $x_1, \ldots, x_N$, we obtain a satisfying assignment of $\phi'_i$.

For the proof in the other direction, if $\phi_i$ is not satisfiable, then for any proof for at least half of all binary strings $r$ the verifier $V$ rejects $\phi_i$. Consider an arbitrary proof and set the variables $x_i$ to the corresponding values from this proof. Observe, that for the binary strings $r$ for which $V$ rejects $\phi_i$ at least one clause in $F_r$ is false. Consequently, at least $\frac{1}{2} \cdot 2^R$ clauses of $\phi'_i$ are false, i.e., a fraction of at least $\frac{1}{2C}$ of the total number of clauses of $\phi'_i$.

Finally, let us put $c = A + 1$, and $r = 1 - \frac{1}{2c}$ and suppose there is a $O^\ast(2^{O\left(\frac{m}{\log^3(m)}\right)})$-time algorithm for Gap E3-SAT.$_{[1-\frac{1}{2c}, 1]}$. Using this algorithm, we can check whether $\phi_i$ is satisfiable in time $2^{O(n)}$ and therefore check whether the initial formula $\phi$ is satisfiable in time $O^\ast(2^{2cR} + 2^{cn} \cdot 2^{O(n)}) = O^\ast(2^C)$. As $c$ is an arbitrary positive number, we obtain a contradiction with the ETH. ⊓⊔

Next, we inspect the chain of three textbook NP-hardness reductions [42], that starts with Gap E3-SAT and ends with Gap MAX CUT. All of them produce an output instance of linear size in terms of the size of the input instance and at the same time preserve gaps. Combining them with the previous theorem, we will establish (almost) subexponential hardness of Gap MAX CUT.

**Theorem 3.2.**

\[
\text{Gap E3-SAT}_{[\alpha, \beta]} \leq_{lin} \text{Gap E4-NAE-SAT}_{[\alpha, \beta]} \leq_{lin} \text{Gap MAX CUT}.
\]
\[
\text{Gap E3-NAE-SAT}_{\frac{1+\epsilon \alpha}{2}, \frac{1+\epsilon \beta}{2}} \leq \frac{\ln p}{\ln p} \text{ Gap Max Cut}_{\frac{1+\epsilon \alpha}{2}, \frac{1+\epsilon \beta}{2}}.
\]

**Proof.** For simplicity, the proof of Theorem 3.2 is split into the following four separate lemmas.

**Lemma 3.3.**
\[
\text{Gap E3-SAT}_{[\alpha, \beta]} \leq \frac{\ln p}{\ln p} \text{ Gap E4-NAE-SAT}_{\{\alpha, \beta\}}.
\]

**Proof.** Given an E3-CNF formula \( \phi = C_1 \land \ldots \land C_m \), we create a new E4-CNF formula \( \phi' = C'_1 \land \ldots \land C'_m \) by adding a new variable \( z \) to each clause. That is, if \( C_i = l_1 \lor l_2 \lor l_3 \), then we set \( C'_i = l_1 \lor l_2 \lor l_3 \lor z \).

If some assignment satisfies \( k \) clauses of \( \phi \), then by additionally setting \( z = 0 \) the corresponding \( k \) clauses of \( \phi' \) are NAE-satisfied. In the other direction, if some assignment \( \phi \) NAE-satisfies \( k \) clauses of \( \phi' \), then its negation \( \bar{\phi} \) also NAE-satisfies \( k \) clauses of \( \phi' \). W.l.o.g. assume that \( \phi \) sets \( z \) to false, which means that \( \phi \) restricted to the variables of \( \phi' \) satisfies \( k \) clauses of \( \phi' \). Consequently, we obtain a gap preserving reduction and the theorem follows.

**Lemma 3.4.**
\[
\text{Gap E4-NAE-SAT}_{\{\alpha, \beta\}} \leq \frac{\ln p}{\ln p} \text{ Gap E3-NAE-SAT}_{\{\alpha, \beta\}}.
\]

**Proof.** Given an E4-CNF formula \( \phi = C_1 \land \ldots \land C_m \), we add \( m \) new variables \( z_1, \ldots, z_m \), and replace every clause \( C_i = l_1 \lor l_2 \lor l_3 \lor l_4 \) with the following two clauses: \( C'_i = l_1 \lor l_2 \lor z_i \), \( C''_i = l_3 \lor l_4 \lor \lnot z_i \), obtaining an E3-CNF formula \( \phi' \). In the following, we show that one can NAE-satisfy at least \( k \) clauses of \( \phi \) if and only if one can NAE-satisfy at least \( m + k \) clauses of \( \phi' \), which suffices to prove the theorem.

First, observe that if an assignment NAE-satisfies \( C_i \), then by setting \( z_i \) appropriately both \( C'_i \) and \( C''_i \) become NAE-satisfied. However, if an assignment does not NAE-satisfy \( C_i \), then setting \( z_i \) to an arbitrary value NAE-satisfies exactly one clause out of \( C'_i \) and \( C''_i \). Consequently, if there is an assignment, which NAE-satisfies \( k \) clauses of \( \phi \), then it can be extended to an assignment of variables of \( \phi' \), which NAE-satisfies \( m + k \) variables of \( \phi' \).

Let us assume that there is an assignment \( \phi \), which NAE-satisfies at least \( m + k \) clauses of \( \phi' \). Note that there is a set \( I \) of at least \( k \) indices \( i \), such that \( \phi \) NAE-satisfies both \( C'_i \) and \( C''_i \). Consider a fixed \( i \in I \) and w.l.o.g. (by symmetry) assume that \( \phi \) sets \( z_i \) to false. As \( \phi \) NAE-satisfies both \( C'_i \) and \( C''_i \), we infer that \( \phi \) sets at least one of the literals \( l_1, l_2 \) to true, and at least one of the literals \( l_3, l_4 \) to false. Consequently, \( \phi \) restricted to the variables of \( \phi' \) NAE-satisfies all the clauses \( C_i \) for \( i \in I \), which finishes the proof of the theorem as \( |I| \geq k \).

**Lemma 3.5.**
\[
\text{Gap E3-NAE-SAT}_{\{\alpha, \beta\}} \leq \frac{\ln p}{\ln p} \text{ Gap Multigraph Max Cut}_{\{\frac{\ln 2 + \epsilon_1}{2}, \frac{\ln 2 + \epsilon_2}{2}\}}.
\]

**Proof.** Given an E3-CNF formula \( \phi = C_1 \land \ldots \land C_m \), let \( n_i \) be the number of occurrences of \( x_i \) in \( \phi \), both in the positive and negative form. We construct a multigraph \( G \) with \( 2n \) vertices and \( 6m \) edges as follows. For each variable \( x_i \), we create two vertices corresponding to literals \( x_i, \lnot x_i \), and add exactly \( n_i \) edges between them. Finally, for every clause \( C_i \), we add a triangle connecting the vertices corresponding to its literals. We will prove that there is an assignment that NAE-satisfies at least \( k \) clauses of \( \phi \) if and only if \( G \) admits a cut of size at least \( 3m + 2k \).

Let us assume that there is an assignment that NAE-satisfies \( k \) clauses of \( \phi \). We put vertices of \( G \) that correspond to literals evaluated to true, on one side of the cut, and remaining literals to the other side of the cut. Every edge connecting \( x_i \) and \( \lnot x_i \) is clearly in the cut, and all of them contribute \( \sum_i n_i = 3m \) to the cut. Furthermore, every triangle corresponding to a NAE-satisfied
clause has exactly two edges cut, all of them contribute 2k to the cut. Thus, we obtain the cut of size at least 3m + 2k.

Let us assume that G has a cut of size at least 3m + 2k. First, suppose that xᵢ and ¬xᵢ are on the same side of the cut for some variable xᵢ. They contribute at most 2nᵢ edges to the cut, and one of them must contribute at most nᵢ edges. We can move it to the other side of the cut without decreasing its size. Thus, we can assume that variables are separated from their negations. The edges connecting xᵢ and ¬xᵢ contribute exactly 3m to the cut. If a clause triangle is cut 2 times, then it corresponds to a NAE-satisfied clause; otherwise, the clause is not NAE-satisfied as all its literals have the same value. Thus, we deduce that the considered cut corresponds to an assignment that NAE-satisfies k clauses of φ.

**Lemma 3.6.**

\[ \text{Gap Multigraph Max Cut}^{[\alpha, \beta]} \preceq_{\text{lin}}^P \text{Gap Max Cut}^{[\frac{\beta}{\alpha}, \frac{\alpha + \beta}{\alpha + \beta}]} \]

**Proof.** Given a multigraph G, for every edge e = uv of G, we create two auxiliary vertices wₑ, zₑ, add three edges uwₑ, wₑzₑ, zₑv, and remove the edge uv, obtaining a resulting graph G'. We claim that G has a cut of size at least k if and only if G' has a cut of size at least 2m + k.

Let C be a cut of G of size k. If e = uv ∉ C, then we put wₑ and zₑ to the side of the cut opposite to u. This way two of the new edges, i.e., uwₑ and zₑv, are in the cut. If e = uv ∈ C, then putting wₑ, zₑ to the opposite sides of u, v, respectively, gives three edges in the cut. All together, we obtain a cut of G' of size 2m + k.

Let C be a cut of G' of size 2m + k. Clearly, there are at least k edges uv of the original graph G, such that all the three edges uwₑ, wₑzₑ, and zₑv are in the cut C. Note that for such an edge uv the vertex wₑ is on the side opposite to u, zₑ is on the side opposite to wₑ and v is on side opposite to zₑ, which means that v is on the side of the cut opposite to u. Consequently, a restriction of C to vertices of G gives a cut of size k.

Lemmas 3.3, 3.4, 3.5, and 3.6 together prove Theorem 3.2.

Thus, we infer:

**Theorem 3.7.** Unless ETH fails, there exist c ∈ N, and 0 ≤ α < β ≤ 1 such that there is no \( 2^{O\left(\frac{m}{\log^c m}\right)} \)-time algorithm for Gap Max Cut\(^{[\alpha, \beta]}\).

Finally, we modify the reduction by Garey et al. [27] from Max Cut to OLA. The construction from Reference [27] introduces a huge clique to the complement of the Max Cut instance; this clique is supposed to separate in the ordering the sides of an optimum max-cut solution. The clique has to be large enough so that the cost of any edge going over the clique eclipses the “noise” contribution that is given by internal ordering of parts on the left and on the right. For this reason the clique is chosen to be of size \( \Theta(n^4) \), which imposes a significant blow-up in the instance size and immediately prevents us from obtaining a desired subexponential hardness result.

This is precisely the moment where we exploit that we are working with an instance of Max Cut with a gap, as starting with a gap instance, we can accommodate the noise in the gap, and therefore we only need the clique to be of linear size. Note that in the following theorem the number of vertices produced is linear; however, the produced instance might be dense, hence we cannot use the notation \( \preceq_{\text{lin}}^P \) and formulate the reduction properties explicitly.

**Theorem 3.8.** There is a polynomial time algorithm, which given an instance I of Gap Max Cut\(^{[\alpha, \beta]}\) with n vertices produces an instance I' of OLA with \( O(n) \) vertices, such that if I admits a cut with at least βm edges, then I' is a yes-instance, and if I does not admit a cut with at least αm edges, then I' is a no-instance.
Proof. Given the input graph $G$, we create a corresponding instance of OLA, $G'$ as follows. We take the complement of $G$, i.e., $\overline{G}$ and add a disjoint clique $C$ of size $Mn$, for $M = \lceil \frac{2}{\beta - \alpha} \rceil$ and fully connect it to $\overline{G}$, creating $G'$.

We will show that $G$ has a cut of size at least $\beta m$ if and only if $G'$ has a linear arrangement of cost at most $\left( \frac{(M+1)n+1}{3} \right) - \beta m \cdot Mn$, which will prove the equivalence of the instances.

Let $(A, B)$ be a cut of size at least $\beta m$. We define a linear arrangement $\pi$ as follows. First, we list the vertices of $A$, then we list the vertices of the clique $C$ and then the vertices of $B$. The order of vertices inside $A, B$ and $C$ can be arbitrary.

Observe that for a clique with $(M+1)n$ vertices, all its linear arrangements have cost exactly $\left( \frac{(M+1)n+1}{3} \right)$. Note that by construction of $G'$, the graph $G'$ together with the edges of $G$ forms a clique with $(M+1)n$ vertices. We can therefore count the cost of a linear arrangement of the union of $G'$ and $G$, which is a clique, as follows:

$$\sum_{u \not\in E(G')} |\pi(u) - \pi(v)| + \sum_{u \in E(G)} |\pi(u) - \pi(v)| = \left( \frac{(M+1)n+1}{3} \right) \tag{*}.$$

Now, the costs of each edge going across the clique, i.e., each edge in $E_{G'}(A, B)$, is at least $Mn$; thus, $\sum_{u \not\in E(G')} |\pi(u) - \pi(v)| \geq \beta m \cdot Mn$, and it implies the desired inequality.

In the other direction, let $\pi$ be a linear arrangement of $G'$ of cost at most $\left( \frac{(M+1)n+1}{3} \right) - \beta m \cdot Mn$. First, we will prove that there exists an optimum linear arrangement $\pi$, such that the vertices of $C$ belong to the same consecutive block of $\pi$. Before that, we will introduce notation regarding to linear arrangements, also used in other proofs.

We assume the vertices of $G'$ are ordered from left to right according to a linear arrangement $\pi$. When we speak about the $i$th vertex (from the left), we mean the vertex mapped to the number $i$ by $\pi$. A set of vertices $U$ is consecutive in $\pi$, if $\pi(U) = \{p, p + 1, \ldots, q - 1, q\}$ for $p, q \in \mathbb{N}$. The set of all vertices that are to the left of every vertex from some set $U$ is called vertices to the left of $U$ and denoted by $L(U)$. Similarly, we define vertices to the right of $U$ and denote them by $R(U)$. A block of $U$ is any inclusion-wise maximal non-empty subset of $U$ that is consecutive in $\pi$. The left-most block of $U$ is the block of $U$ whose vertices are mapped to the smallest values by $\pi$. The second left-most block of $U$ the first block of $U$ to the right of the left-most block of $U$. The inner block of $U$ is the set of all vertices from $V(G') \setminus U$ located simultaneously to the right of the left-most block of $U$ and to the left of the second left-most block of $U$ (in the case when $U$ forms a single block, the inner block does not exist).

Claim 3.9. There exists an optimum linear arrangement $\pi$ of $G'$ such that the vertices of $C$ are consecutive in $\pi$.

Proof. Let us choose $\pi$ to be an optimum linear arrangement of $G'$ that minimize the number of vertices of $G$ that lie between the vertices of $C$, i.e., $f(\pi) = |\{v \in V(G) : \pi(u) < \pi(v) < \pi(w), u, w \in V(C)\}|$ is the smallest possible. We claim that the vertices of $C$ are consecutive in $\pi$, i.e., $f(\pi) = 0$.

Assume that this is not the case. Let $X$ be the inner block of $C$, and we consider the following two cases:

- $|E_G(L(X) \cap V(G), X)| \leq |E_G(X, R(X) \cap V(G))|,$
- $|E_G(L(X) \cap V(G), X)| > |E_G(X, R(X) \cap V(G))|.$

In each of them, we will create another linear arrangement that will contradict the assumptions we made.
If the first case occurs, then let $C'$ be the left-most block of $C$. We swap places of $C'$ and $X$. Then, we consider how it affects the cost of $\pi$. First, we note that by (*), the optimality of $\pi$ can be equivalently restated as maximizing the sum of cost of edges of $G$, i.e., $\pi$ is optimal if $\sum_{u,v \in E(G)} |\pi(u) - \pi(v)|$ is the largest possible. We claim that the value of this sum has not decreased. Clearly, the cost of edges with both endpoints in $V(G) \setminus X$ or $X$ has not changed. Hence, it is enough to inspect the edges with exactly one endpoint in $X$. Thus, we are left with edges going from $X$ to $L(X) \cap V(G)$ or from $X$ to $R(X) \cap V(G)$. An edge going from $X$ to $L(X) \cap V(G)$ has decreased its length by $|C'|$, and an edge going from $X$ to $R(X) \cap V(G)$ has increased its length by $|C'|$. Thus, as $|E_G(L(X) \cap V(G), X)| \leq |E_G(X, R(X) \cap V(G))|$ implies, the overall contribution of those edges has not decreased, but the number of vertices between $C$ has dropped by $|X|$, leading to the contradiction.

If the second case occurs, then let $C'$ be the second-left most block of $C$. Similarly, we swap places of $C'$ and $X$. Again, only affected edges are the ones that have exactly one endpoint in $X$. The inequality $|E_G(L(X), X)| > |E_G(X, R(X))|$ implies that $\sum_{u,v \in E(G)} |\pi(u) - \pi(v)|$ has increased at least by $|X|$, contradicting the optimality of $\pi$.

Thus, we have proved that $C$ has to be consecutive in $\pi$. □

Let $\pi$ be the optimum linear arrangement from the claim above. We define a cut $(A, B)$, by taking $A$ to be $L(C)$ and $B$ to be $R(C)$. Then, by (*), we know

$$\beta m \cdot Mn \leq \sum_{u \in E(G)} |\pi(u) - \pi(v)|$$

$$= \sum_{u \in E(G), \ u \in A} |\pi(u) - \pi(v)| + \sum_{u \in E(G), \ u \in B} |\pi(u) - \pi(v)| + \sum_{u \in E(G), \ u \in A, v \in B} |\pi(u) - \pi(v)|$$

$$< nm + (Mn + n) \cdot |E_G(A, B)| \leq 2nm + Mn \cdot |E_G(A, B)|$$

Thus,

$$|E_G(A, B)| > \beta m - \frac{2m}{M} \geq \beta m - (\beta - \alpha) \cdot m = \alpha m.$$  

By the promise given by the gap problem, we know that if $G$ admits a cut greater than $\alpha m$, then it actually admits a cut of size at least $\beta m$.

The proof of Theorem 1.5 follows immediately from Theorems 3.7 and 3.8.

### 4 Sparse Reduction

We now introduce a polynomial-time Turing reduction from a gap version of the Min Bisection problem on $d$-regular graphs to Optimum Linear Arrangement. Its key property is that the instances of the former problem result in instances of Optimum Linear Arrangement with linear
number of vertices and bounded degree. Even though the created instance of OPTIMUM LINEAR ARRANGEMENT is a multigraph, it does not cause any additional difficulties in further reductions described in Section 5. The reduction allows one to distinguish between instances of MIN BISECTION with at most \( \alpha m \) edges and at least \( \beta m \) edges in the optimum cut for some fixed choice of \( 0 \leq \alpha < \beta \leq 1 \) by solving the resulting instance of OPTIMUM LINEAR ARRANGEMENT. This relates OPTIMUM LINEAR ARRANGEMENT to the following hypothesis:

**Hypothesis 1.2.** There exist \( 0 \leq \alpha < \beta \leq 1 \), and an integer \( d > \frac{4}{\beta - \alpha} \), such that there is no \( 2^{o(n)} \)-time algorithm for GAP MIN BISECTION\((d)\) \([\alpha, \beta]\).

The main result of this section is:

**Theorem 1.6.** Unless Hypothesis 1.2 fails, there exists an integer \( d \in \mathbb{N} \) such that there is no \( 2^{o(n)} \)-time algorithm solving the decision version of OLA\(_{\leq} (d)\) in multigraphs.

We first describe a transformation \( T(\cdot) \) from an instance \( G \) of MIN BISECTION to an instance of OPTIMUM LINEAR ARRANGEMENT that forms the key component of our reduction. Then, we introduce several technical claims about its properties. Finally, we prove Theorem 1.6 by showing how to decide the instances of GAP MIN BISECTION based on the cost of the optimum arrangement of \( T(G) \). The fact that our reduction exhibits only a linear increase in the size of the instance is crucial in achieving the \( 2^{\Omega(n)} \) bound.

The result of transformation \( T(\cdot) \) is influenced by several parameters. The choice of their values is deferred to the proof of Theorem 1.6. Consider an instance \( G \) of the MIN BISECTION problem, where \( G \) is a \( d_G \)-regular graph. Assume \( V(G) = \{v_1, \ldots, v_n\} \). The transformation produces a graph \( G' := T(G) \) with the vertex set \( \{v_1, \ldots, v_n, x_1, \ldots, x_{Z \cdot \lceil \phi n \rceil}\} \), where \( Z \in \mathbb{N} \) and \( \phi \in (0, 1) \) are constants chosen later.

Note that \( G' \) contains the vertices of \( G \). Indeed, we are going to construct the edge-set in such a way that \( G \) is an induced subgraph of \( G' \). It is actually convenient to introduce notation for some of the induced subgraphs of \( G' \). The subgraph with the vertex set \( \{x_1, \ldots, x_{Z \cdot \lceil \phi n \rceil}\} \) is denoted by \( H \). The graph \( H \) is (arbitrarily) divided into \( Z \) disjoint induced subgraphs \( H_i \) of size \( \lceil \phi n \rceil \) each, for some constants \( Z \in \mathbb{N} \) and \( \phi \in (0, 1) \).

The result of the transformation is illustrated in Figure 2. The edge-set of \( G' \) is constructed as follows:

- The induced subgraph of \( G' \) on \( \{v_1, \ldots, v_n\} \) is \( G \).
- We construct a \( d_H \)-regular expander \( G_{\lfloor H \rfloor, d_H} \) using Theorem 2.2 satisfying \( h(G_{\lfloor H \rfloor, d_H}) \geq \rho_H \), and add its edges on the vertices of \( H \) (the value \( \rho_H \) will be determined later).

\[ G \]

\[ H \approx G_{Z \cdot \lceil \phi n \rceil, d_H} \]

\[ G' \]

Fig. 2. The resulting instance \( G' = T(G) \) after applying the reduction. The original graph \( G \) is an induced subgraph of \( G' \) with its edges shown in green. The edges of the bipartite graph added between \( V(G) \) and \( V(H_1) \) are shown in blue.
For each $i \in \{1, \ldots, Z\}$, we construct a $d_{H_i}$-regular expander $G|_{H_i \setminus d_{H_i}}$ using Theorem 2.2 satisfying $h(G|_{H_i \setminus d_{H_i}}) \geq p_{H_i}$, and add its edges on the vertices of $H_i$ (the value $p_{H_i}$ will be determined later).

For each $i \in \{1, \ldots, Z\}$, we add a bipartite graph on parts $V(G)$ and $V(H_i)$ such that all vertices of $V(G)$ have degree 1 in this bipartite graph and the degrees of vertices from $V(H_i)$ differ by at most 1. We denote the maximum degree of the $V(H_i)$ part of this added bipartite graph by $\Delta_{H_i,G}$. It is at most $\lceil \frac{1}{p_{H_i}} \rceil$.

Note that we are constructing a multigraph, that is when an edge is to be added several times in the construction process, we keep all its copies.

In the following part, we give the proof of correctness of the transformation, as well as determine the parameters driving the reduction. In the proof, we first show that the vertices of $H$ have to be consecutive in an optimum linear arrangement. Next, we show that actually vertices of each small expander $H_i$ are consecutive in such an ordering. This is crucial when we analyze the change of the cost of the ordering when moving a vertex of $G$ from one side of $H$ to the other, and in turn prove that in an optimum ordering the parts of $G$ to the left and to the right of $H$ are almost of the same size. Interestingly, in our reduction, we have to use the hypothetical oracle solving the decision version of Optimum Linear Arrangement to find the cost of an optimum ordering of an auxiliary graph by using binary search.

The constructed $G'$ is influenced (apart from the input graph $G$) by our choice of parameters $Z$, $\varphi$, $p_{H_i}$, and $p_{H_i}$, which in turn influence $d_H$ and $d_{H_i}$ by Theorem 2.2. The lemmas below impose a particular structure on the optimum linear arrangement of $G'$, provided certain inequalities between these parameters are satisfied. Eventually, the lemmas are employed in the proof of the main theorem of this section.

The following technical Swapping Lemma establishes a condition on degrees in two consecutive vertex sets of $G'$ under which the swapping of the two sets results in a decreased cost of the ordering.

**Lemma 4.1 (Swapping Lemma).** Consider an ordering $\pi$ of any finite graph $G'$. Assume that the sets $X, Y \subseteq V(G')$ are consecutive and $X$ immediately precedes $Y$. Let $L := L(X)$ and $R := R(Y)$. Assume

- the value $P_X$ upper bounds the degree of vertices from $X$ in the induced bipartite subgraph $G'[L,X]$,
- $P_C$ is an upper bound on the maximum degree of $G'[X,Y]$, and
- $P_Y$ is an upper bound on the degree of a vertex from $Y$ in $G'[Y,R]$.

Finally, let $p$ be a lower bound on the average degree of a vertex from $X$ in $G'[X,R]$. Then the inequality $p > P_X + 2P_C + P_Y$ implies that swapping the vertices of $X$ with the vertices of $Y$ in the order specified by $\pi$ results in a decrease in the cost of the ordering.

**Proof.** The situation is illustrated in Figure 3. The length of all edges connecting a pair of vertices from one of the sets $L, X, Y, R$ remains unchanged after swapping $X$ and $Y$ in the ordering. The same holds for edges connecting $L$ with $R$. The length of each edge connecting $X$ and $Y$ increases by at most $|X| + |Y| \leq 2 \max\{|X|, |Y|\}$. The cost of each edge connecting $X$ and $L$ increases by at most $|Y|$. Similarly, the cost of each edge connecting $Y$ and $R$ increases by at most $|X|$. However, the edges connecting $X$ and $R$ are shortened, each by $|Y|$. The upper bounds on maximum degrees and the lower bound on average degree from the statement of the lemma allow us to lower bound the decrease in the cost of the ordering after the swap is performed. For example, the decrease in total cost of the edges connecting $X$ with $R$ is at least $p|X||Y|$. The decrease in cost after swapping...
The vertex set of \( G' \) is partitioned into four sets, \( L, X, Y, \) and \( R \), in Lemma 4.1. The bounds \( p_X, p_Y, p_C \) (upper bounds), and \( p \) (a lower bound) on the degrees of the induced bipartite subgraphs are also shown. The bound is at least

\[
p|X||Y| - 2 \min\{|X|, |Y|\} p_C \max\{|X|, |Y|\} - |X|p_X|Y| - |Y|p_Y|X|,
\]

which is equal to

\[
|X||Y|(p - 2p_C - p_X - p_Y).
\]

Assuming the inequality from the lemma, this is strictly larger than zero. □

We now make several claims about the optimum ordering \( \pi \) of \( G' := T(G) \), where \( G \) is a \( d_G \)-regular graph. Recall that \( Z, \varphi, p_H, p_{H_i}, d_H, \) and \( d_{H_i} \) are the parameters of the transformation \( T(\cdot) \) still to be determined. We assume, that \( p_H > 3 \Delta_{H,G} + 3Z + d_G, p_{H_i} > d_{H_{i+1}} + 4d_H + 2\Delta_{H,G} \), for all \( i \in \{1, \ldots, Z - 1\} \) (**).

**Lemma 4.2.** If \( \pi \) is an optimum linear arrangement of \( G' \), then \( V(H) \) is consecutive in \( \pi \).

**Proof.** Suppose \( V(H) \) is not consecutive in \( \pi \). Consider the left-most block of \( V(H) \) and denote its elements by \( X \). We can assume that \( |X| \leq \frac{|H|}{2} \) — otherwise, we take the right-most block of \( V(H) \) and proceed with a mirrored version of the following argument. Denote by \( Y \) the inner block of \( V(H) \) and set \( L := L(X), R := R(Y) \). The following choice of values satisfies the assumptions on degree upper-bounds of Lemma 4.1:

\[
P_X := \Delta_{H,G}, \quad P_Y := d_G + Z, \quad P_C := \Delta_{H,G} + Z.
\]

Since \( H \) is an expander, \( |X| \leq |H|/2 \) and \( H \setminus X \subseteq R \), we take \( p = p_H \). It remains to show the inequality from the statement of the Swapping Lemma. We have

\[
p = p_H > 3\Delta_{H,G} + 3Z + d_G = p_X + 2p_C + p_Y.
\]

Thus, we can swap \( X \) and \( Y \) and decrease the cost of the ordering. This contradicts the optimality of \( \pi \). □

**Lemma 4.3.** Let \( i \in \{1, \ldots, Z\} \). If \( \pi \) is an optimum linear arrangement of \( G' \), and for each \( i' < i \) the vertices of \( H_{i'} \) are consecutive in \( \pi \), then the vertices of \( H_i \) are consecutive in \( \pi \). (For the purposes of this lemma, we set \( d_{H_{Z+1}} := 0 \).)

Moreover, for each \( i \in \{1, \ldots, Z\} \) the vertices of \( H_i \) are consecutive in any optimum linear arrangement of \( H \).

**Proof.** First, we argue about the consecutiveness of \( H_i \) in an ordering of \( G' \). Assume \( H_i \) not to be consecutive in an optimum linear arrangement of \( G' \) and denote by \( X \) the left-most block of \( V(H_i) \) in \( \pi \). Similarly to the situation in the proof of Lemma 4.2, we can assume \( |X| \leq |H_i|/2 \).
(Otherwise, we consider the right-most block instead and proceed with a mirrored version of the argument.) We show that moving $X$ to the right decreases the cost of the arrangement.

Denote by $u$ the vertex positioned by $\pi$ immediately to the right of $X$. Due to Lemma 4.2, we know $u \notin V(G)$. Therefore, $u \in V(H_j)$ for $j \neq i$. We distinguish two cases: either $j < i$ or $j > i$.

Suppose that $u \in V(H_i)$ for $j < i$. Note that $H_j$ is consecutive in $\pi$. We set $Y := V(H_j)$, $L := L(X)$, and $R := R(Y)$. Again, we employ the Swapping Lemma. The following degree upper-bounds satisfy its assumptions:

$$P_X := \Delta_{H,G} + d_H, \quad P_Y := \Delta_{H,G} + d_H, \quad P_C := d_H.$$

Since $H_i$ is an expander, $|X| \leq |H_i|/2$ and $V(H_i) \setminus X \subseteq R$, we can set the average degree lower-bound $p$ to $p_{H_i}$. By the inequality $p_{H_i} > d_{H_{i+1}} + 4d_H + 2\Delta_{H,G}$ from (**), we have

$$p = p_{H_i} > 4d_H + 2\Delta_{H,G} = P_X + 2P_C + P_Y.$$

Thus, the inequality from the Swapping Lemma holds, and we can use it to decrease the cost of ordering, contradicting the optimality of $\pi$.

Suppose therefore that $u \in V(H_j)$ for $j > i$. We now use Lemma 4.1 again to move the block $X$ one position to the right, effectively swapping $X$ and $Y := \{u\}$. We set $L := L(X)$, $R := R(Y)$. This time, we set

$$P_X := \Delta_{H,G} + d_H, \quad P_Y := \Delta_{H,G} + d_H + d_{H_{i+1}}, \quad P_C := d_H.$$

Similarly to the previous cases, we set $p := p_{H_i}$. The inequality from the Swapping Lemma is again satisfied:

$$p = p_{H_i} > 4d_H + 2\Delta_{H,G} + d_{H_{i+1}} = P_X + 2P_C + P_Y.$$

Once more, we get a contradiction with the optimality of $\pi$.

Observe that when we consider an optimum arrangement of $H$, we can use exactly the same arguments by substituting $\Delta_{H,G} = 0$. This proves that the vertices of $H_i$ are consecutive in any optimum arrangement of $H$, which proves the second part of the lemma.

Due to Lemma 4.2, we know that an optimum linear arrangement of $G'$ places vertices of $H$ consecutively, assuming the inequalities from (***) are satisfied. Furthermore, by iterating Lemma 4.3, we get that within $H$, the vertices of each $H_i$ are grouped together in the arrangement. To be precise, in the optimum ordering the subgraphs $H_i$ are placed in the order $H_{k_1}, H_{k_2}, \ldots, H_{k_Z}$, where $(k_1, k_2, \ldots, k_Z)$ is a permutation of $\{1, 2, \ldots, Z\}$. The vertices of $G$ can thus be only to the left of $H$ or to its right. The next lemma shows that $H$ divides the graph $G$ into two roughly equal parts.

**Lemma 4.4.** Assume $G'$ has been constructed by the transformation $T(\cdot)$ with parameters satisfying the inequalities (***) and $Z \geq 2$. Moreover, assume $\gamma = 3\beta d_G$. Consider an optimum linear arrangement $\pi$ of $G'$ and set $A := L(H), B := R(H)$. Then, $||A| - |B|| \leq \gamma n$.

**Proof.** Assume the imbalance $||A| - |B||$ is strictly bigger than $\gamma n$. Without loss of generality, assume $|A| > |B|$. We consider the vertex $u$ such that $\pi(u) = 1$ (i.e., the one placed on the left side of the arrangement).

Moving $u$ to the right-most position results in the following changes in the cost of the arrangement. The cost associated with the edges of $G$ might be increased by at most $d_G(Z \varphi n) + n \leq d_G(Z \varphi n + Z + n)$. Note that for sufficiently large $n$, we have $\frac{Z \varphi n}{2} \geq Z$ and by the assumption of the Lemma we have $\frac{Z \varphi n}{2} \geq n$, therefore the cost associated with the edges of $G$ increases by at most $d_G(Z \varphi n + Z + n) \leq 2d_GZ \varphi n$. 
In addition to this, the vertex $u$ is connected to precisely one vertex $v_i$ of each $H_k$. Before moving $u$, the edge cost of $uv_i$ was $|A| + (i - 1)[\varphi n] + r_i$, for some $0 \leq r_i \leq [\varphi n] - 1$ and after, it becomes $|B| + ((Z - i + 1)[\varphi n] - r_i)$, thus the contribution of all those edges has changed by $$((|B| - |A|)Z + \sum_{i=1}^{Z}(Z - i + 1)[\varphi n] - r_i) - \sum_{i=1}^{Z}((i - 1)[\varphi n] + r_i) = (|B| - |A|)Z + \sum_{i=1}^{Z}([\varphi n] - 2r_i) < \gamma nZ + Z[\varphi n] + \sum_{i=1}^{Z}([\varphi n] - 2r_i) < 2d_GZ[\varphi n] + Z[\varphi n].$$

Note that in the last inequality, we have used the assumption $[\varphi n] < d_G[\varphi n]$, which holds for $d_G \geq 2$ and sufficiently large $n$, as $\varphi$ is a constant.

Combining this cost change with the cost increase of edges of $G$ being at most $2d_GZ[\varphi n]$, we obtain that moving $u$ to the right-most position causes the decrease in the cost of $\pi$, leading to a contradiction. 

Being equipped with all the required tools, we are ready to prove Theorem 1.6.

**Proof of Theorem 1.6** We prove the theorem by introducing a Turing reduction from $\text{Gap Min Bisection}(d)_{[\alpha, \beta]}$ to $\text{OLA}_{\leq}(d')$, where

$$d > \frac{4}{\beta - \alpha},$$

and $d'$ is some constant depending on $d$, $\alpha$, and $\beta$. The reduction proceeds in the following way.

Given an instance $G$ of $\text{Gap Min Bisection}(d)_{[\alpha, \beta]}$, we create an equivalent instance $(G', k)$ of $\text{OLA}_{\leq}(d')$ based on the transformation $T(\cdot)$ applied to $G$. As mentioned in the beginning, the reduction is a Turing reduction. It is therefore equipped with an oracle solving the decision version of OLA. First, we use this oracle to compute the cost of an optimum linear arrangement of $H$, (we denote it by $\text{OLA}(H)$), by employing binary search. We do it as the value $k$ will depend on $\text{OLA}(H)$, among other parameters.

We start by establishing the parameters of the transformation in the following order.

- First, we set $\gamma := \frac{\beta - \alpha}{4}$ and $\varphi := \frac{\gamma}{3d_G}$. Note that $\gamma, \varphi \in (0, 1)$ and $\gamma, \varphi$ satisfy the condition $\gamma = 3\varphi d_G$ from Lemma 4.4.
- Next, we set the value of $Z$ to the following integer:

$$Z := \frac{2(2\alpha + 1)}{(\beta - \alpha)\varphi}.$$ 

In particular $Z \geq \frac{2}{\varphi}$ satisfying the condition from Lemma 4.4. Moreover, the following inequality holds:

$$2(2\alpha + 1) \leq (\beta - \alpha)Z\varphi. \quad (2)$$

- By construction $\Delta_{H,G} \leq \lceil \frac{\varphi}{\gamma} \rceil = \lceil \frac{3d_G}{\gamma} \rceil$.
- Next, we define $p_H$, which in turn determines the value of $d_H$ by Theorem 2.2:

$$p_H := 3\Delta_{H,G} + 3Z + d_G + 1.$$ 

The additive term +1 is just to make sure the inequality from (**) is strict and it could be replaced by any positive constant.
- Finally, we set the values $p_{H_i}$:

$$p_{H_i} := d_{H_{i+1}} + 4d_H + 2\Delta_{H,G} + 1.$$
in the decreasing order \( i = Z, \ldots, 1 \), where for simplicity we denote \( d_{H_{Z+1}} = 0 \). Note that the value of \( p_{H_i} \) determines the value of \( d_{H_i} \) via Theorem 2.2.

Observe that with the above definition of all the constants, we satisfy all the requirements of (***) and Lemmas 4.2, 4.3, and 4.4.

Let \( G' \) be the result of the transformation \( T(G) \) with the above choice of parameters. The value \( k \) is then set as follows:

\[
k := \text{OLA}(H) + am \cdot (Z[\phi n] + n) + m \cdot \frac{n}{2} + \left( \frac{n}{2} + 1 \right) \frac{n}{2} Z + n \sum_{i=1}^{Z} i[\phi n].
\]

(3)

It remains to show that \( G \) has a bisection of size at most \( am \) if and only if \( G' \) has an optimum linear arrangement of cost at most \( k \).

Let us assume \( G \) has a bisection with at most \( am \) edges. We claim that \( k \) is an upper bound on the cost of an optimum linear arrangement of \( G' \). This is because it accounts for all costs associated with an ordering of \( G' \) constructed from the optimum bisection of \( G \). Denote by \( A, B \) the partition of \( V(G) \) corresponding to an optimum bisection of \( G \). We create an ordering \( \pi \) that first lists all vertices of \( A \), then the vertices of \( H \) in the order of an optimum linear arrangement of \( H \), and finally the vertices of \( B \).

The first term of Equation (3) is the cost of all edges inside \( H \). The second term upper bounds the cost of edges of \( G \) between \( A \) and \( B \): there are at most \( am \) of them, and we upper bound the cost of every such edge by \( |V(G')| = (Z[\phi n] + n) \). In the third term, we account for the cost of edges within \( A \) and within \( B \). There are at most \( m \) of them, and since \( |A| = |B| = \frac{n}{2} \), every such edge has cost at most \( \frac{n}{2} \). The last term is an upper bound on the cost of edges connecting \( G \) and \( H \).

By Lemma 4.3 the optimum ordering of \( H \) has the form \( H_{k_1}, H_{k_2}, \ldots, H_{k_2} \). Every vertex \( v \) of \( G \) has an edge to exactly one vertex of each \( H_{k_j} \). If \( v \in A \), then we may bound its cost by \( j(v) + i[\phi n] \), where \( j(v) \) is the length of the part of the ordering from \( v \) to the first vertex in \( H \). We first count the contribution of the \( j(v) \)-terms in the above expression for all choices of \( v \in A \). Since \( |A| = n/2 \), summing over all \( v \in A \) and \( i = 1, \ldots, Z \) we get \( \sum_{j=1}^{|A|} j Z = \left( \frac{n}{2} + 1 \right) \frac{n}{2} Z \). The situation is analogous for \( B \). The last term of (3) is obtained by summing the remaining edge costs \( i[\phi n] \) for all \( i = 1, \ldots, Z \) and \( v \in V(G) \). This proves the claimed upper bound.

In the other direction, we start with assuming that the graph \( G' \) has an optimum arrangement \( \pi \) of cost at most \( k \). The aim is to prove that \( G \) has a bisection of size at most \( am \). Lemma 4.2, Lemma 4.3, and Lemma 4.4 together impose a certain structure on \( \pi \). Particularly, the vertices of \( H \) are placed together in \( \pi \). We use this fact to construct a bisection of \( G \). Set \( A := L(H) \) and \( B := R(H) \), and if \( |A| < |B| \), then replace \( A \) with \( B \). Note that \( (A, B) \) is a partition of \( V(G) \) and these two sets might have different sizes with the imbalance bounded by Lemma 4.4. We now bound the number of edges between \( A \) and \( B \). To this end, we lower bound the cost of \( \pi \) in terms of \( |E_G(A, B)| \).

Specifically, it is at least

\[
\text{OLA}(H) + |E_G(A, B)| \cdot Z[\phi n] + \left( \frac{n}{2} + 1 \right) \frac{n}{2} Z + n \sum_{i=1}^{Z} (i - 1)[\phi n].
\]

(4)

There, the first term of Equation (4) accounts for the cost of the edges of \( H \), which is lower bounded by \( \text{OLA}(H) \). The second term of Equation (4) is a lower bound on the cost of edges of \( G \) going across the partition \( (A, B) \). There are \( |E_G(A, B)| \) of such edges and each contributes at least \( Z[\phi n] \) to the cost. The third term lower bounds the cost of edges connecting \( G \) to \( H \). Every vertex \( v \) of \( G \) has an edge to exactly one vertex from \( H_{k_i} \), where \( k_i \) is the index such that vertices of \( H_{k_i} \) occupy the \( i \)-block of \([\phi n]\) positions among \( V(H) \). Similarly to the analysis above, we lower bound the cost of such an edge by \( j(v) + (i - 1)[\phi n] \). The contribution of \( j(v)'s \) from the above expression
for all choices of \( v \in G \) is equal to \((\sum_{j=1}^{\lfloor |A|/2 \rfloor} j + \sum_{j=1}^{\lfloor |B|/2 \rfloor} j)Z \geq (2 \sum_{j=1}^{\lfloor n/2 \rfloor} j)Z \geq (n^2 + 1)\frac{n}{Z}. \) The remaining part is obtained by summing \((i - 1)[\varphi n]\) over all \( v \in G \) and \( i = 1, \ldots, Z \). Comparing Equation (3) with Equation (4), we obtain

\[
|E_G' (A, B)| \cdot Z[\varphi n] \leq \alpha m \cdot (Z[\varphi n] + n) + m \cdot n \frac{Z}{4} + n \sum_{i=1}^{Z} i[\varphi n] - n \sum_{i=1}^{Z} (i - 1)[\varphi n]
\]

\[
= \alpha m \cdot Z[\varphi n] + \frac{2(\alpha + 1)mn}{4} + n \sum_{i=1}^{Z} [\varphi n] \leq \alpha m \cdot Z[\varphi n] + \frac{(\beta - \alpha)m \cdot Z[\varphi n]}{4} + n \cdot Z[\varphi n] \varphi n \leq [\varphi n]
\]

We infer that \(|E_G'(A, B)| < \frac{\alpha + 3\beta}{4} m. \) Next, we create a bisection \((A', B')\) of \( G \) as follows. Let \( C \) be a set of \( \frac{|A| - |B|}{2} \) arbitrary vertices of \( A \), then we put \( A' = A - C, B' = B \cup C \). By moving these vertices, we get at most \( \frac{\gamma n}{2} \cdot d_G = \gamma m \) additional edges in the cut; thus,

\[
|E_G'' (A', B')| \leq |E_G'(A, B)| + \gamma m < \frac{\alpha + 3\beta}{4} m + \frac{\beta - \alpha}{4} m = \beta m.
\]

Therefore, we have \(|E_G'' (A', B')| < \beta m. \) Since the problem considered is a gap problem, we know that \( G \) admits either a bisection of size at most \( am \) or there is no bisection of size less than \( \beta m \). Therefore, we conclude the instance admits a bisection of size at most \( am. \) \( \square \)

5 LOWER BOUNDS FOR MINIMUM FILL-IN AND OTHER COMPLETION PROBLEMS

In this section, we prove Theorems 1.1 and 1.3, that is prove conditional lower bounds (under ETH and under Hypothesis 1.2) for parameterized completion problems such as MINIMUM FILL-IN, CHAIN COMPLETION, PROPER INTERVAL COMPLETION, INTERVAL COMPLETION, THRESHOLD COMPLETION, and TRIVIALLY PERFECT COMPLETION. As a starting point, we use Theorems 1.5 and 1.6; hence, our goal is to transform an instance of OLA into an instance of a graph completion problem. The main reduction of this section, which transforms an instance of OLA to CHAIN COMPLETION is a slight modification of the reduction of Yannakakis [44], with the only difference being that on a bounded degree instance of OLA, we obtain a linear number of vertices in the final instance of CHAIN COMPLETION. This fact is crucial to prove Theorem 1.3 while for proving Theorem 1.1 it is enough to use the original version of Yannakakis’ reduction.

**Definition 5.1.** A bipartite graph \((A, B, F)\) with vertices \( A \uplus B \) and edges \( F \) is a chain graph if the set of vertices \( A \) (called left side) can be ordered \( v_1, v_2, \ldots, v_n \) (called left order) in such a way that \( N(v_i) \subseteq N(v_j) \subseteq \cdots \subseteq N(v_n). \)

In the CHAIN COMPLETION problem given a bipartite graph \((A, B, F)\) one is asked to add a minimum number of edges \( F' \subseteq A \times B \setminus F \) such that \((A, B, F \cup F')\) is a chain graph.

**Lemma 5.2.** There is a polynomial time algorithm, which given an instance \( I = (G = (V, E), k) \) of OPTIMUM LINEAR ARRANGEMENT creates an equivalent instance \( I' = (G' = (A, B, F), k') \) of CHAIN COMPLETION, such that the number of vertices of \( G' \) is bounded by \( O(\Delta_G \cdot |V|) \), where \( \Delta_G \) is the maximum degree of \( G \). The reduction works even if \( G \) is a multigraph.

**Proof.** As the left side of \( G' \), we take \( A = V \). For each vertex \( v \in V \) create a set of \( \Delta_G \) new vertices \( S_v = \{v_e : e \in \delta_G(v)\} \cup \{v_i : \deg(v) < i \leq \Delta_G\}. \) We define \( B \) as the union of all the sets \( S_v \), thus \( B \) contains exactly \( \Delta_G \cdot |V| \) vertices. The set of edges \( F \) is constructed as follows. For each
Fig. 4. Transformation of Optimum Linear Arrangement to Chain Completion.

For a given ordering $\pi$ of the vertices of the graph $G = (V, E)$ denote by $C(G, \pi)$ the cost of arrangement induced by this ordering (i.e., $C(\pi, G) = \sum_{uv \in E} |\pi(u) - \pi(v)|$). For an ordering $\sigma$ of the left side $A$ of the bipartite graph $G' = (A, B, F)$ denote by $E(G', \sigma)$ the number of edges that we should add to obtain a minimal chain graph in which the left order coincides with $\sigma$. We will prove the following claim.

**Claim 5.3.** For any ordering $\pi$ of $V$ (or equivalently $A$), we have $E(G', \pi) = C(G, \pi) + \Delta_G \frac{n(n-1)}{2} - 2|E|$.

**Proof.** Let us inspect what is the number of edges in a minimal chain bipartite graph $G''$, which has left order $\pi = (v_1, v_2, \ldots, v_n)$ and contains $G' = (A, B, F)$ as a subgraph. We know that $N(v_i) \subseteq N(v_j)$ for any $i < j$, which means that each $x \in S_{v_i}$ must be connected to all vertices from the set $\{v_i, v_{i+1}, \ldots, v_n\}$. So it means that each vertex from $S_{v_i}$ is connected to at least $(n+1) - i$ vertices from the set $A$. Moreover, by minimality of $G''$ the vertex $x$ is connected to exactly this number of the vertices if $x$ does not correspond to any edge in $G$, i.e., when $x$ is of degree exactly one in $G$. If a vertex $w_i \in S_{v_i}$ corresponds to some edge $e \in G$ with endpoints $v_i, v_j$ (note there could be several edges with equal endpoints as we are working with multigraphs), then the vertex $w_i$ is connected to vertices $v_i, v_j$ in the graph $G'$. Hence, in a minimal chain graph $G''$ the degree of $w_i$ is either $(n+1) - i$ or $(n+1) - j = (n+1) - i + (i - j)$ depending on whether $i < j$ or $i > j$. Note that there is a second vertex $w_j \in S_{v_j}$, which also corresponds to the edge $e$. The degrees of $w_i$ and $w_j$ in $G''$ both are equal to $(n+1) - i$ or $(n+1) - j$, depending whether $i < j$ or $i > j$. In both cases the sum of degrees $w_i, w_j$ in $G''$ can be written as $((n+1) - i) + ((n+1) - j) + |i - j|$. Hence, for each edge $e$ with endpoints $v_i, v_j$, we have additional cost of $|i - j|$. Summing up, we
infer that the number of edges in $G''$ equals

$$\Delta_G \left( \sum_{i=1}^{n} ((n + 1) - i) \right) + \sum_{i,j \in E} |i - j| = \Delta_G \frac{n(n + 1)}{2} + C(G, \pi).$$

The number of added edges equals the number of edges in $G''$ minus the number of edges in $G'$. So, we add exactly

$$\left( \Delta_G \frac{n(n + 1)}{2} + C(G, \pi) \right) - (\Delta_G n + 2|E|) = \Delta_G \frac{n(n - 1)}{2} + C(G, \pi) - 2|E|$$

edges.

Equivalence of the instances $I$ and $I'$ follows from the claim, and it proves the lemma.

Having an instance of Chain Completion we transform it further to an instance of other completion problems by simply making $A$ a clique, or by making both $A$ and $B$ cliques. By inspecting the forbidden subgraphs definition of each graph class we infer the equivalence of the instances, which is enough to prove Theorems 1.1 and 1.3.

Classes of chordal, interval, proper interval, threshold, and trivially perfect graphs have many characterizations. For our purposes the most convenient one is by the set of forbidden induced subgraphs [12]. The characterization is presented in Table 1.
Table 1. Characterization of Graph Classes by Forbidden Induced Subgraphs

| Graph class name | Forbidden induced subgraphs |
|------------------|-----------------------------|
| Chordal          | $C_n$ for $n \geq 4$        |
| Proper Interval  | claw, 2-net, 3-tent, $C_n$ for $n \geq 4$ |
| Interval         | bipartite claw, umbrella, $n$-net for $n \geq 2$, $n$-tent for $n \geq 3$, $C_n$ for $n \geq 4$ |
| Threshold        | $2K_2$, $C_4$, $P_4$        |
| Trivially perfect| $C_4$, $P_4$                |

**Lemma 5.4.** There are polynomial time reductions from the Chain Completion problem to the Minimum Fill-In, Interval Completion, Proper Interval Completion, Threshold Completion, and Trivially Perfect Completion problems and these reductions do not change the vertex set.

**Proof.** For any bipartite graph $H' = (U_1, U_2, F)$ consider a graph $Ch(H') = (U_1, U_2, F \cup \{uv|u, v \in U_1\} \cup \{uv|u, v \in U_2\})$. In Reference [44] it is shown that any bipartite graph $H'$ is a chain graph if and only if $Ch(H')$ is a chordal graph. It means that to reduce an instance of Chain Completion $H = (A, B, F)$ to Minimum Fill-In it is enough to construct cliques on sets of vertices $A$ and $B$. So the constructed graph is a union of two cliques and some edges between cliques. Hence, any arbitrarily completion of this graph does not contain a claw, bipartite claw, umbrella, $p$-net, $q$-tent for $p \geq 2, q \geq 3$ as these graphs have an independent set of size 3 and a union of two cliques does not. It follows that solutions for Minimum Fill-In, Proper Interval Completion, Interval Completion problems on such instances coincide, and we can look at this reduction to Minimum Fill-In as a reduction to Proper Interval Completion or Interval Completion problems.

It is left to show a reduction from a Chain Completion instance to Threshold Completion and Trivially Perfect Completion instances. Having a Chain Completion instance with a bipartite graph $H = (A, B, F)$, we consider Trivially Perfect Completion and Threshold Completion problems on the graph $G = (A \cup B, F \cup \{(u, v)|u, v \in A\})$. We just add edges such that $A$ becomes a clique. We show that a minimum chain completion of the graph $H$ corresponds to a completion towards trivially perfect or threshold graph. Let $F'$ be a solution of Chain Completion for the graph $H$. Consider graph $G' = (A \cup B, F \cup \{(u, v)|u, v \in A\} \cup F')$, $G'$ is a union of independent set and clique plus some edges between independent set and clique. So it does not contain induced $2K_2$ or $C_4$. If $G'$ contains induced $P_4 = v_1v_2v_3v_4$, then $v_2, v_3$ belong to clique and $v_1, v_4$ to independent set. However, this contradicts the fact that edges between clique and independent set form a chain completion. As $G'$ does not contain induced $2K_2, P_4, C_4$, it is trivially perfect and threshold graph. Let now $F'$ denote solution of Trivially Perfect Completion or Threshold Completion on instance $G'$. To finish the proof of correctness of reduction it is enough to show that $(A, B, (F \cup F') \cap E(A, B))$ is a chain graph. If this graph is not a chain graph, then it must contain two independent edges $v_1v_2, v_3v_4$ [44]. However, in such case graph on vertices $v_1, v_2, v_3, v_4$ will induce a $P_4$ in graph $G' \cup F'$, which contradict the fact that $G' \cup F'$ is a threshold or trivially perfect graph.

To sum up all our reductions from Chain Completion to Minimum Fill-In, Proper Interval Completion, Interval Completion, Threshold Completion, and Trivially Perfect Completion add some edges to a graph of Chain Completion instance and do not change a vertex set of a graph.

At this point, we almost proved Theorems 1.1 and 1.3.

**Theorem 1.1.** Unless ETH fails, there is an integer $c \geq 1$ such that there are no $2^{O(\sqrt{n}/\log^c n)}$, and consequently no $2^{O(k^{1/4}/\log^c k)} \cdot n^{O(1)}$ algorithms for the following problems: Minimum Fill-In,
proof. If the statement is not true, then for some of the problems there is an algorithm running in time $2^{O(\sqrt{n}/\log^c n)}$, for every constant c. Having an instance of Optimum Linear Arrangement on $n$ vertices, we can reduce it to problem under consideration with $(\Delta_G + 1)n = O(n^3)$ vertices by Lemmas 5.2 and 5.4. This gives us an $2^{O(\sqrt{\Delta}/\log^c n^2)} = 2^{O(n/\log^c n)}$ time algorithm that contradicts Theorem 1.5. As $k \leq n^2$, we also have a $2^{O(\sqrt{n}/\log^c k)} \cdot n^{O(1)}$ lower bound on the running time. □

Theorem 1.3. Unless Hypothesis 1.2 fails, there is no $2^{O(n+m)}$-time algorithm for Chain Completion, and there are no $2^{\omega(n)}$-time algorithms for Minimum Fill-In, Interval Completion, Proper Interval Completion, Trivially Perfect Completion, and Threshold Completion. Consequently, none of these problems can be solved in time $2^{O(\sqrt{\Delta})} \cdot n^{O(1)}$.

proof. In Section 4, we transformed a $d$-regular Min Bisection instance to a Optimum Linear Arrangement instance with bounded degree. Pipelined with Lemma 5.2, we get a reduction from a $d$-regular Min Bisection instance to a Chain Completion instance with $O(n)$ vertices and edges. So $2^{O(n+m)}$-time algorithm for Chain Completion contradicts Hypothesis 1.2. By Lemma 5.4, we can reduce a Chain Completion instance to Minimum Fill-In, Interval Completion, Proper Interval Completion, Trivially Perfect Completion, and Threshold Completion instance without changing the vertex set. Combining all the three reductions in one, we get the reductions from Min Bisection to Minimum Fill-In, Proper Interval Completion, Interval Completion, Threshold Completion, and Trivially Perfect Completion, which transform an instance with $n$ vertices into an instance with $O(n)$ vertices. This leads to a $2^{\Omega(n)}$ lower bound for all the discussed completion problems as well as to a $2^{O(\sqrt{\Delta})} \cdot n^{O(1)}$ lower bound, because $k \leq n^2$. □

6 HARDNESS OF FEEDBACK ARC SET IN TOURNAMENTs

In this section, we prove Theorem 1.4, that is, the lower bound on the complexity of Feedback Arc Set in Tournaments. We start with preparing an appropriately hard instance of Feedback Arc Set in general digraphs, so that we can apply the reduction of Ailon et al. [1].

6.1 Preparing a Hard Instance of FAS

By (E$_3$$_1$1,E$_2$$_2$,$d$)-SAT, we denote the version of 3-Sat where every variable has

- exactly 1 positive occurrence in a clause of size 3,
- exactly 1 negative occurrence in a clause of size 3,
- exactly $d$ positive occurrences in clauses of size 2,
- exactly $d$ negative occurrences in clauses of size 2, and
- there are no clauses of size 1.

Similarly as before, Gap (E$_3$$_1$1,E$_2$$_2$,$d$)-SAT$_{[\alpha, \beta]}$ for $0 \leq \alpha < \beta \leq 1$ is the problem of distinguishing whether the maximum number of clauses that can be satisfied in a given instance of (E$_3$$_1$1,E$_2$$_2$,$d$)-SAT is at most $\alpha m$, or at least $\beta m$, where $m$ is the total number of clauses. We now give a hardness result for Gap (E$_3$$_1$1,E$_2$$_2$,$d$)-SAT$_{[\alpha, \beta]}$.

Lemma 6.1. There exists a positive integer $d$ such that

$$\text{Gap E}_3\text{-NAE-SAT}_{[\alpha, 1]} \leq_P \text{Gap (E}_3\text{-E}_2\text{-d)-SAT}_{[\alpha, \beta]}$$

proof. Let $\phi$ be the input E3-CNF formula. For every variable $x$, let $n(x)$ be the number of occurrences of $x$ in $\phi$. Let $G_x$ be a $d$-regular expander with $n(x)$ vertices, obtained using Theorem 2.2 for $p = 2$; suppose vertices of $G_x$ are numbered 1, 2, ..., $n(x)$. Create $\psi$ as follows:
Replace $x$ with $n(x)$ new variables $x_1, x_2, \ldots, x_{n(x)}$, corresponding to the occurrences of $x$ in $\phi$;

- For every $ij \in E(G_\phi)$ with $i \neq j$, introduce two clauses $(\neg x_i \lor x_j)$ and $(x_i \lor \neg x_j)$, which (if unbroken) force the evaluation of $x_i$ to be equal to that of $x_j$;

- For every $ii \in E(G_\phi)$, introduce a trivial clause $(\neg x_i \lor x_i)$;

- Perform the same construction for every other variable of $\phi$;

- For every 3-clause $C$ of $\phi$, introduce two new clauses $C'$ and $C''$ to $\psi$. Clause $C'$ is constructed from $C$ by replacing each occurrence of some variable $x$ with the new variable $x_i$ corresponding to this occurrence; the polarity of the literals in $C'$ is the same as in $C$. Clause $C''$ is constructed from $C'$ by reversing the polarity of each literal. For example, if $C = (x \lor \neg y \lor z)$, then $C' = (x_i \lor \neg y_j \lor z_k)$ and $C'' = (\neg x_i \lor y_j \lor \neg z_k)$, where $i, j, k$ are the indices of the occurrences of $x, y, z$ in $C$, respectively.

It is easy to see that every variable of the new formula $\psi$ appears in exactly two 3-clauses, once positively and once negatively. Also, it has exactly $2d$ occurrences in 2-clauses: $d$ positive and $d$ negative. Since the original formula had only clauses of size 3, the new formula has exactly $3m$ variables and $m(2 + 3d)$ clauses.

If there is a variable evaluation $\lambda$ for $\phi$ that NAE-satisfies all clauses of $\phi$, then we can construct a variable evaluation $\lambda'$ for $\psi$ by assigning all the variables $x_i$ that originate in variables $x$ the value $\lambda(x)$. Then it is easy to see that $\lambda'$ satisfies all the clauses of $\psi$.

Suppose now that every variable evaluation for $\phi$ NAE-satisfies at most $am$ clauses, and for the sake of contradiction suppose that there is a variable evaluation $\lambda'$ for $\psi$ that satisfies more than $(1 + \alpha + 3d)m$ clauses. Let $x$ be a variable of $\phi$, and let us modify $\lambda'$ as follows: assign all the variables $x_1, x_2, \ldots, x_{n(x)}$ the value that is taken by the majority of these variables in the original evaluation $\lambda'$ (breaking ties arbitrarily). Observe that since $h(G_\phi) \geq 2$, this step cannot decrease the number of satisfied clauses: if $q$ is the number of variables out of $x_1, x_2, \ldots, x_{n(x)}$ that take the majority value, then by replacing their values by the majority value, we can unsatisfy at most $2q$ 3-clauses, but we satisfy at least $h(G_\phi) \cdot q \geq 2q$ 2-clauses that were previously unsatisfied. By performing this operation for every variable of $\phi$, we can assume without loss of generality that in $\lambda'$ all the variables originating in the same variable of $\phi$ are assigned the same value. This naturally defines a variable evaluation $\lambda$ for $\phi$. Then, provided $\lambda'$ satisfied more than $(1 + \alpha + 3d)m$ clauses of $\psi$, we infer that $\lambda$ NAE-satisfies more than $am$ clauses of $\phi$. This is a contradiction. \hfill $\square$

We now turn our attention to the Feedback Vertex Set problem in general directed multigraphs: given a directed multigraph $G$, find the smallest possible subset of vertices $X$ such that $G - X$ is acyclic. By FVS-BAL($d$), we denote the variant of FVS where the input directed multigraph has no loops, and is 2d-regular and balanced, i.e., the indegree and the outdegree of every vertex is equal to $d$. Again, GapFVS-BAL($d$)-SAT is the gap problem where we need to distinguish the cases when the optimum size of $X$ is at most $\alpha n$ and at least $\beta n$, where $n$ is the number of vertices of the multigraph.

**Lemma 6.2.**

$$\text{Gap } (E3_{1,1},E2_{d,d})\text{-SAT}_{[\alpha,1]} \leq_{\text{pol}} \text{Gap FVS-BAL}(d + 2)^{1/2/(4-a)/2}.$$

**Proof.** Let $\phi$ be the input instance of $(E3_{1,1},E2_{d,d})\text{-SAT}$, and let $n$ be the number of variables of $\phi$. Then $m$, the number of clauses of $\phi$, is equal to $(2/3 + d)n$.

Construct a directed multigraph $G$ as follows. For every variable $x$ of $\phi$, create two vertices $u^x_\top$ and $u^x_\bot$, corresponding to setting $x$ to true and false, respectively. Add edges $(u^x_\top, u^x_\bot)$ and $(u^x_\bot, u^x_\top)$ to the edge set, for every variable $x$ of $\phi$. Moreover, for every 2-clause of $\phi$ add a 2-cycle between vertices corresponding to its literals (e.g., clause $x \lor \neg y$ gives rise to edges $(u^x_\top, u^y_\bot)$ and $(u^y_\bot, u^x_\top)$).
and similarly for every 3-clause of $\phi$ add a 3-cycle between vertices corresponding to its literals, oriented arbitrarily. Note that in this manner trivial clauses of the form $(\neg x \lor x)$ give rise to additional copies of the 2-cycle $(u^x_+, u^x_-)(u^x_+, u^x_-)$. This concludes the construction of $G$. It is easy to verify using the assumed properties of $\phi$ that every vertex of $G$ has indegree and outdegree equal to $d + 2$, and moreover in the construction, we did not introduce loops. Let $n' = 2n$ be the number of vertices in $G$.

Suppose first that there exists a variable evaluation $\lambda$ for $\phi$ that satisfies all the clauses of $\phi$. Define $X$ to be the set of all the vertices $u^x_{\lambda(x)}$ for $x$ being a variable of $\phi$; note that $|X| = n = n'/2$. Since $\lambda$ satisfies all the clauses, and every variable of $\phi$ participates in exactly one 3-clause positively and in exactly one 3-clause negatively, then it is easy to see that all the weakly connected components of $G - X$ are either isolated vertices or single edges. Thus, $G - X$ is acyclic.

Assume now that every variable evaluation for $\phi$ satisfies at most $am$ clauses, and for the sake of contradiction suppose that there exists a set $X$ with $|X| < \frac{4 - \alpha}{6} n' = \frac{4 - \alpha}{3} n$ such that $G - X$ is acyclic. Observe that from each pair $\{u^x_+, u^x_-\}$ at least one vertex has to belong to $X$. Define a variable evaluation $\lambda$ for $\phi$ as follows: if $|\{u^x_+, u^x_-\} \cap X| = 1$, then $\lambda(x)$ is such that $u^x_{\lambda(x)} \in X$, and otherwise $\lambda(x)$ is chosen arbitrarily. Observe that the first alternative holds for a set of more than $n - \frac{1 - \alpha}{3} n$ variables; let us denote them by $S$. Since $G - X$ is acyclic, each of 2- and 3-cycles constructed for a clause $C$ of $\phi$ has at least one vertex from $X$. If all the variables of $C$ belong to $S$, then it can be easily seen that this implies that $\lambda$ satisfies $C$. Hence, the only clauses of $\phi$ that can be unsatisfied by $\lambda$ are the ones that contain at least one variable outside $S$. Every variable of $\phi$ occurs in at most $2d + 2$ clauses, and there are less than $\frac{1 - \alpha}{3} \cdot n$ variables outside $S$, which means that $\lambda$ unsatisfies less than $\frac{(1 - \alpha)(2d + 2)}{3} \cdot n$ clauses. Hence, the fraction of unsatisfied clauses is less than

$$\frac{(1 - \alpha)(2d + 2) n}{3m} = \frac{(1 - \alpha)(2d + 2)}{(2 + 3d)} \leq 1 - \alpha.$$ 

This is a contradiction. $\square$

Finally, there is a well-known reduction that reduces Feedback Vertex Set to Feedback Arc Set in the directed setting. This reduction preserves the gap. In the following, by $\text{Gap FAS}(d)_{[\alpha, \beta]}$, we denote the problem of determining, for a given directed multigraph $G$ without loops whose underlying undirected multigraph is $d$-regular, whether the minimum number of edges that needs to be removed from $G$ to make it acyclic is at most $am$ or at least $bm$, where $0 \leq \alpha < \beta \leq 1$ and $m$ is the number of edges in $G$.

**Lemma 6.3.**

$$\text{Gap FVS-BAL}(d)_{[\alpha, \beta]} \leq_{\text{lin}}^p \text{Gap FAS}(d + 1)_{\left[\frac{d + \beta}{d + 1}, \frac{d + \alpha}{d + 1}\right]}.$$

**Proof.** Let $G$ be the input directed multigraph, and let $n$ and $m$ denote the numbers of edges and vertices of $G$, respectively; by the assumption that $G$ is $2d$-regular, we know that $m = dn$. Construct a graph $G'$ as follows:

- For every $u \in V(G)$ create two vertices $u^-, u^+ \in V(G')$ and an edge $(u^-, u^+) \in E(G')$;
- For every edge $(u, v) \in E(G)$, create an edge $(u^+, v^-)$.

This concludes the construction of $G'$. Let $E_1, E_2$ be the sets of edges constructed in the first and second bullet point, respectively. Since $G$ was $2d$-regular and balanced, we infer that every vertex $u^+$ has outdegree $d$ and indegree 1, whereas every vertex $u^-$ has outdegree 1 and indegree $d$. Thus, $G'$ has $n' = 2n$ vertices and $m' = m + n = (d + 1)n$ edges.
Suppose first that \( X \) is a subset of vertices of \( G \) with size at most \( \alpha n \) such that \( G - X \) is acyclic. Let \( F = \{(u^+, u^-) \mid u \in X \} \subseteq E_1 \). Then it can be easily seen that \( G' - F \) is acyclic, and \( |F| = |X| \leq \alpha n = \frac{\alpha}{d+1}m' \).

Assume now that every subset \( X \subseteq V(G) \) for which \( G - X \) is acyclic has size at least \( \beta n \), and for the sake of contradiction suppose that there is a set \( F \subseteq E(G') \) such that \( G' - F \) is acyclic and \( |F| < \frac{\beta}{d+1}m' \). Observe that if \( F \) contains some edge \((u^+, v^-)\) in \( E_2 \), then we could modify \( F \) by removing \((u^+, v^-)\) from \( F \) and adding \((v^-, v^+)\) to \( F \) (unless it is not already contained in \( F \), in which case, we do not add any edge to \( F \)). This operation can only decrease the number of edges in \( F \) and preserves the property that \( G' - F \) is acyclic; this is because after removing \((v^-, v^+)\), \( v^- \) becomes a sink. Thus, without loss of generality, we can assume that \( F \subseteq E_1 \). Let \( X \) be the set of vertices \( u \in V(G) \) for which \((u^-, u^+)\) \( \in \) \( F \). Since \( G' - F \) is acyclic, it easily follows that \( G - X \) is also acyclic. Moreover, \( |X| = |F| < \frac{\beta}{d+1}m' = \beta n \). This is a contradiction. \( \square \)

Finally, observe that in an instance of FAS without loops one can subdivide every edge once, which doubles the number of edges while not changing the size of the optimum solution. Thus, application of this reduction to the gap problem shrinks the gap twice and makes the directed graph at hand simple: It has no loops, no parallel edges, and moreover, if \( (u, v) \) is an edge, then \((v, u)\) is not. By combining this observation with Theorem 3.1 and Lemmas 3.3, 3.4, 6.1, 6.2, and 6.3, and Randomized ETH, we obtain the following result.

**Theorem 6.4.** Unless Randomized ETH fails, there exist \( 0 \leq \alpha < \beta \leq 1 \), \( c \geq 1 \), and \( d > 0 \) such that there is no \( 2^{O(n^{\alpha}/m)} \) algorithm for Gap FAS\([\alpha, \beta]\) on directed simple graphs of maximum total degree \( d \).

### 6.2 Reducing FAS to FAST

In this section, we will use the following formulation of the Chernoff bound.

**Theorem 6.5 (Chernoff bound).** Suppose \( X_1, X_2, \ldots, X_r \) are independent random variables with values in \( \{0, 1\} \), and let \( X \) be their sum, then
\[
\Pr(|X - \mathbb{E}X| \geq \delta \mathbb{E}X) \leq 2 \exp \left(-\frac{\delta^2 \mathbb{E}X}{3}\right) \quad \text{for} \quad \delta \leq 1.
\]

**Theorem 1.4.** Unless Randomized ETH fails, there is an integer \( c \geq 1 \) such that there is no \( 2^{O(\sqrt{n}/\log^c n)} \), and consequently no \( 2^{O(\frac{k^{1/4}}{\log^c k})} \cdot n^{O(1)} \) algorithm for Feedback Arc Set in Tournaments.

**Proof.** We provide a randomized reduction that essentially reiterates the argument of Ailon et al. [1]. In the analysis, we use the known fact that for any directed graph \( H \), \( \text{fas}(H) \) is equal to the minimum possible number of edges oriented backwards (called feedback edges) in an ordering of vertices of \( V(H) \), where \( \text{fas}(H) \) is the optimum size of a feedback arc set in \( H \). Thus, we may equivalently think of the FAS problem as finding an ordering \( \pi \) of \( V(H) \) that minimizes the number of feedback edges. For an ordering \( \pi \) of \( V(H) \), by \( \text{fas}(H, \pi) \), we denote the number of feedback edges in the ordering \( \pi \).

Let \( \alpha, \beta, c, d \) be the constants given by Theorem 6.4, and let \( G \) be an instance of Gap FAS\([\alpha, \beta]\), where \( G \) is a simple directed graph with \( n \) vertices and \( m \) edges and has maximum total degree \( d \). Note that, w.l.o.g., we may assume \( m \geq n \), as otherwise there is a vertex in \( G \) with no outgoing edges, which can be safely removed. Let us fix an integer \( k \), to be determined later. We consider the \( k \)-blow up \( G_k \) defined as follows: For every \( u \in V(G) \), we create \( k \) vertices \( u_1, u_2, \ldots, u_k \) in \( G_k \), and for all \( 1 \leq i, j \leq k \), we put \((u_i, u_j) \in E(G_k)\) if and only if \((u, v) \in E(G)\). Thus, vertices \( u_i \) are twins. Ailon et al. [1], based on a communication by Alon, argue that there is an optimum ordering for \( G_k \),
which may be obtained by taking an optimum ordering for $G$ and replacing every vertex $u \in V(G)$ by a block consisting of vertices $\{u_i\}_{1 \leq i \leq k}$ in any order; hence, in particular, fas$(G_k) = k^2 \cdot fas(G)$. For an ordering $\sigma$ of $V(G)$, let $\sigma_k$ be an ordering of $V(G_k)$ constructed in the manner described above.

Construct a tournament $T_k$ from $G_k$ by adding edges between every pair of vertices that are not connected by an edge in $G_k$, where the orientations of these edges are chosen independently and uniformly at random. Observe that $|E(G_k)| = k^2 \cdot |E(G)| \leq dk^2n/2$. Let $R_k = (V(T_k), E(T_k) \setminus E(G_k))$ be the directed graph consisting only of the edges picked at random. Then, for a sufficiently large $n$, we have that $|E(R_k)| = \left\{ \left( \frac{n_k}{2} \right) - |E(G_k)| \right\} \geq \left( \frac{n_k}{4} \right)^2$, and of course $|E(R_k)| \leq \left( \frac{n_k}{2} \right)^2$.

We now prove that with high probability, FeedbackArcSet($T_k$) is closely related to FeedbackArcSet($G_k$), because the number of feedback edges that need to be chosen from the edges picked at random is concentrated around the expected value.

Let us fix some ordering $\pi$ of $V(G_k)$, then fas($T_k, \pi)$ = fas($G_k, \pi$) + fas($R_k, \pi$). For $e \in E(R_k)$, let $X_e$ be the indicator random variable having value 1 if $e$ is a feedback edge w.r.t. $\pi$, and 0 otherwise. Let also $X = \sum_{e \in E(R_k)} X_e$; then $E(X) = |E(R_k)|/2$. Let $\eta = (2 - \alpha)/3$. Since $X_e$'s are independent, from the Chernoff bound (by putting $\delta = \frac{n_k^2}{EX}$), we obtain that

$$\Pr(|X - EX| \geq \eta k^2 n) \leq 2 \exp\left(-\frac{2\eta^2 k^4 n^2}{3|E(R_k)|}\right) \leq 2 \exp(-4\eta^2 \eta^2 k^2/3).$$

Suppose now that there exists an ordering $\pi$ of $V(G)$ that has at most $am$ feedback edges. Then, with probability at least $1 - 2 \exp(-4\eta^2 k^2/3)$, we have that

$$\text{fas}(T_k, \pi_k) \leq ak^2 m + |E(R_k)|/2 + \eta k^2 n \leq \frac{2\alpha + \beta}{3} \cdot k^2 m + |E(R_k)|/2. \quad (5)$$

Hence, if we can set $k$ to be large enough, then conclusion Equation (5) holds with probability at least $3/4$.

Suppose now that fas$(G) \geq \beta m$. Then, for a fixed ordering $\sigma$ of $V(G_k)$, we have that with probability at least $1 - 2 \exp(-4\eta^2 k^2/3)$ it holds that

$$\text{fas}(T_k, \sigma) \geq \text{fas}(G_k, \sigma) + |E(R_k)|/2 - \eta k^2 n \geq k^2 \text{fas}(G) + |E(R_k)|/2 - \eta k^2 n$$

$$\geq k^2 \beta m + |E(R_k)|/2 - \eta k^2 n \geq \frac{\alpha + 2\beta}{3} \cdot k^2 m + |E(R_k)|/2. \quad (6)$$

We would like to infer that with probability $\geq \frac{3}{4}$ this conclusion holds for all the possible orderings $\sigma$, and for this, we will use the union bound. Observe that the number of orderings $\sigma$ of $V(G_k)$ is $(nk)! = \exp(O(nk \log(nk)))$, while the probability of failure for each of them is at most $2 \exp(-4\eta^2 k^2/3)$. Since $\eta$ is a positive constant, simple computations show that by setting $k = \Theta(n \log n)$, we have that $(nk)! \cdot 2 \exp(-4\eta^2 k^2/3) \leq 1/4$, and hence, conclusion Equation (6) holds simultaneously for all orderings $\sigma$ with probability at least $3/4$.

Suppose now that FeedbackArcSet in Tournaments admitted an algorithm with running time $2^{O(n^{1/2}/\log^c n)}$ for $c' = c + \frac{1}{2}$, where $c$ is as in Theorem 6.4. Apply this algorithm to the constructed tournament $T_k$ to compute fas($T_k$). In case fas$(G) \leq am$, then with probability at least $3/4$, we have that fas($T_k$) $\leq 2\frac{\alpha + \beta}{3} \cdot k^2 m + |E(R_k)|/2$. In case fas$(G) \geq \beta m$, then with probability at least $3/4$, we have that fas($T_k$) $\geq \frac{\alpha + 2\beta}{3} \cdot k^2 m + |E(R_k)|/2$. Since $2\frac{\alpha + \beta}{3} < \frac{\alpha + 2\beta}{3}$, these two alternatives are disjoint and the algorithm can, with double-sided error, resolve the input instance of GapFAS$[\alpha, \beta]$. Since $|V(T_k)| = \Theta(n^2 \log n)$, this procedure runs in time $2^{O(\frac{n^2}{\log^c n})}$. Together with Theorem 6.4, it implies that ETH fails.
7 CONCLUSIONS

In this work, we have given evidence that $2^{O(\sqrt{k} \cdot \text{polylog}(k))} \cdot n^{O(1)}$ can be the final answer for the running times of parameterized algorithms for Minimum Fill-In, Interval Completion, Proper Interval Completion, Trivially Perfect Completion, Threshold Completion, and Chain Completion. This evidence is based on a new complexity hypothesis connected to the hardness of approximation for the Min Bisection problem. Thus, the answer given by us is not completely satisfactory: The lower bounds that we can give only under the assumption of ETH are much weaker. Rather, our results uncover a surprising link between the parameterized algorithms for Minimum Fill-In and related problems, and the approximability of Min Bisection. Thus, it seems that the question about the optimality of the former has a much deeper, fundamental nature.

Therefore, we believe that our work strongly motivates further investigation of Hypothesis 1.2. Can this conjecture be linked to ETH and possibly some other strong conjectures like SETH, the existence of linear PCPs, or the conjectures proposed by Feige [22]? Or maybe it can be simply disproved?

Our improved lower bound for Feedback Arc Set in Tournaments still has a gap between $k^{1/4}$ and $k^{1/2}$. A closer inspection of the proof uncovers a fundamental obstacle for why we cannot achieve tightness: the Chernoff concentration bound used in the proof of Theorem 1.4 is essentially tight, because in every tournament on $n$ vertices there is a feedback arc set of size $(\binom{n}{2})^2 - \Omega(n^{3/2})$ [29]. If this error term was of magnitude $\Theta(n)$ instead of $\Theta(n^{3/2})$, then our approach would give a tight result for FAST. Can this problem be circumvented, or maybe the high anticoncentration of the number of feedback edges in a random ordering of a tournament can be exploited algorithmically to obtain a faster algorithm?

Finally, even assuming Hypothesis 1.2, we do not get tight bounds, due to the (poly)logarithmic factors in the exponent describing the running time of the existing algorithms for completion problems. Bridging this gap can be seen as another forthcoming goal.

We note that after the first version of this work was published, Cao and Sandeep [13] have shown a tight lower bound for the Minimum Fill-In problem under the ETH via a direct and novel reduction from the Vertex Cover problem.

ACKNOWLEDGMENTS

We thank Per Austrin for valuable discussions about different versions of the PCP theorem and anonymous reviewers for their helpful comments.

REFERENCES

[1] Nir Ailon, Moses Charikar, and Alantha Newman. 2008. Aggregating inconsistent information: Ranking and clustering. J. ACM 55, 5 (2008).
[2] Noga Alon. 2006. Ranking tournaments. SIAM J. Discrete Math. 20, 1 (2006), 137–142.
[3] Noga Alon, Daniel Lokshpanov, and Saket Saurabh. 2009. Fast FAST. In Proceedings of the 36th International Colloquium on Automata, Languages, and Programming (ICALP’09) (Lecture Notes in Computer Science), Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas (Eds.), Vol. 5555. Springer, 49–58.
[4] Sanjeev Arora and Boaz Barak. 2009. Computational Complexity—A Modern Approach. Cambridge University Press. Retrieved from http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264.
[5] Piotr Berman and Marek Karpinski. 2002. Approximation hardness of bounded degree MIN-CSP and MIN-BISECTION. In Proceedings of the 29th International Colloquium on Automata, Languages, and Programming (ICALP’02) (Lecture Notes in Computer Science), Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephane Eidenbenz, and Ricardo Conejo (Eds.), Vol. 2380. Springer, 623–632.
[6] Ivan Bliznets, Fedor V. Fomin, Marcin Pilipczuk, and Michal Pilipczuk. 2014. A subexponential parameterized algorithm for proper interval completion. In Proceedings of the 22nd Annual European Symposium on Algorithms (ESA’14) (Lecture Notes in Computer Science), Andreas S. Schulz and Dorothea Wagner (Eds.), Vol. 8737. Springer, 173–184.

ACM Transactions on Algorithms, Vol. 16, No. 2, Article 25. Publication date: March 2020.
[7] Ivan Bliznets, Fedor V. Fomin, Marcin Pilipczuk, and Michal Pilipczuk. 2018. Subexponential parameterized algorithm for internal completion. ACM Trans. Algor. 14, 3 (2018), 35.
[8] Edouard Bonnet, Bruno Escoffier, Eun Jung Kim, and Vangelis Th. Paschos. 2015. On subexponential and FPT-time inapproximability. Algorithmica 71, 3 (2015), 541–565.
[9] Edouard Bonnet, Michael Lampis, and Vangelis Th. Paschos. 2015. Time-approximation trade-offs for inapproximable problems. CoRR abs/1502.05828 (2015).
[10] Edouard Bonnet and Vangelis Th. Paschos. 2014. Parameterized (in)approximability of subset problems. Oper. Res. Lett. 42, 3 (2014), 222–225.
[11] Edouard Bonnet and Vangelis Th. Paschos. 2014. Sparsification and subexponential approximation. CoRR abs/1402.2843 (2014).
[12] Andreas Brandstädt, Jeremy P. Spinrad et al. 1999. Graph Classes: A Survey. Vol.3. Siam.
[13] Yixin Cao and R. B. Sandeep. 2017. Minimum fill-in: Inapproximability and almost tight lower bounds. In Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA’17). Society for Industrial and Applied Mathematics, Philadelphia, PA, 875–880. Retrieved from http://dl.acm.org/citation.cfm?id=3039686.3039741.
[14] Pierre Charbit, Stéphan Thomassé, and Anders Yeo. 2007. The minimum feedback arc set problem is NP-hard for tournaments. Combinat. Probabil. Comput. 16, 1 (2007), 1–4.
[15] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. Parameterized Algorithms. Springer.
[16] Timothy A. Davis. [n.d.]. Fill-Reducing Orderings. SIAM, 99–134.
[17] Erik D. Demaine and MohammadTaghi Hajiaghayi. 2008. The bidimensionality theory and its algorithmic applications. Comput. J. 51, 3 (2008), 292–302.
[18] Irit Dinur. 2007. The PCP theorem by gap amplification. J. ACM 54, 3 (2007), 12.
[19] Pål Gronås Drange, Markus Sortland Dregi, Daniel Lokshtanov, and Blair D. Sullivan. 2015. On the threshold of intractability. In Proceedings of the Annual European Symposium on Algorithms (ESA’15) (Lecture Notes in Computer Science), Vol. 9294. Springer, Berlin, 411–423.
[20] Pål Gronås Drange and Michal Pilipczuk. 2015. A polynomial kernel for trivially perfect editing. In Proceedings of the Annual European Symposium on Algorithms (ESA’15) (Lecture Notes in Computer Science), Vol. 9294. Springer, Berlin, 424–436.
[21] Uriel Feige. 2002. Relations between average case complexity and approximation complexity. In Proceedings of the 34th Annual ACM Symposium on Theory of Computing, John H. Reif (Ed.). ACM, 534–543.
[22] Uriel Feige. 2009. Faster FAST (feedback arc set in tournaments). CoRR abs/0911.5094 (2009).
[23] R. Ravi, David S. Johnson, and Lan Lan. 1995. Fast approximation schemes for matroid and knapsack problems. SIAM J. Comput. 25, 5 (1995), 984–1006.
[24] Subhash Khot and Igor Shinkar. 2015. On hardness of approximating the parameterized clique problem. Electr. Colloq. Comput. Complex. 22 (2015), 13.
Lower Bounds for the Parameterized Complexity of Minimum Fill-in

[35] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. 2011. Lower bounds based on the exponential time hypothesis. Bull. EATCS 105 (2011), 41–72.

[36] A. Lubotzky, R. Phillips, and P. Sarnak. 1988. Ramanujan graphs. Combinatorica 8, 3 (1988), 261–277. DOI: https://doi.org/10.1007/BF02126799

[37] Dániel Marx. 2007. On the optimality of planar and geometric approximation schemes. In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS’07). IEEE Computer Society, 338–348.

[38] Dániel Marx. 2012. What’s next? Future directions in parameterized complexity. In The Multivariate Algorithmic Revolution and Beyond—Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday (Lecture Notes in Computer Science), Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx (Eds.), Vol. 7370. Springer, 469–496.

[39] Moshe Morgenstern. 1994. Existence and explicit constructions of q+ 1 regular Ramanujan graphs for every prime power q. J. Combinator. Theory, Ser. B 62, 1 (1994), 44–62.

[40] Harald Räcke. 2008. Optimal hierarchical decompositions for congestion minimization in networks. In Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Cynthia Dwork (Ed.). ACM, 255–264.

[41] J. D. Rose. 1972. A Graph-theoretic Study of the Numerical Solution of Sparse Positive Definite Systems of Linear Equations. Academic Press, New York.

[42] Thomas J. Schaefer. 1978. The complexity of satisfiability problems. In Proceedings of the 10th Annual ACM Symposium on Theory of Computing. ACM, 216–226.

[43] Yngve Villanger, Pinar Heggernes, Christophe Paul, and Jan Arne Telle. 2009. Interval completion is fixed parameter tractable. SIAM J. Comput. 38, 5 (2009), 2007–2020.

[44] Mihalis Yannakakis. 1981. Computing the minimum fill-in is NP-complete. SIAM J. Alg. Discrete Methods 2, 1 (1981), 77–79. DOI: https://doi.org/10.1137/0602010

Received May 2016; revised July 2019; accepted December 2019