Edge Clique Partition and Cover Beyond Independence

Fedor V. Fomin
University of Bergen, Norway

Petr A. Golovach
University of Bergen, Norway

Danil Sagunov
Saint Petersburg State University, Russia

Kirill Simonov
University of Bergen, Norway

Abstract

Covering and partitioning the edges of a graph into cliques are classical problems at the intersection of combinatorial optimization and graph theory, having been studied through a range of algorithmic and complexity-theoretic lenses. Despite the well-known fixed-parameter tractability of these problems when parameterized by the total number of cliques, such a parameterization often fails to be meaningful for sparse graphs. In many real-world instances, on the other hand, the minimum number of cliques in an edge cover or partition can be very close to the size of a maximum independent set $\alpha(G)$.

Motivated by this observation, we investigate *above-*α parameterizations of the edge clique cover and partition problems. Concretely, we introduce and study EDGE CLIQUE COVER ABOVE INDEPENDENT SET (ECC/α) and EDGE CLIQUE PARTITION ABOVE INDEPENDENT SET (ECP/α), where the goal is to cover or partition all edges of a graph using at most $\alpha(G) + k$ cliques, and k is the parameter. Our main results reveal a distinct complexity landscape for the two variants. We show that ECP/α is fixed-parameter tractable, whereas ECC/α is NP-complete for all $k \geq 2$, yet can be solved in polynomial time for $k \in \{0, 1\}$. These findings highlight intriguing differences between the two problems when viewed through the lens of parameterization above a natural lower bound.

Finally, we demonstrate that ECC/α becomes fixed-parameter tractable when parameterized by $k + \omega(G)$, where $\omega(G)$ is the size of a maximum clique of the graph G. This result is particularly relevant for sparse graphs, in which ω is typically small. For H-minor free graphs, we design a subexponential algorithm of running time $f(H)^{\sqrt{k}} \cdot n^{\mathcal{O}(1)}$.

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms; Theory of computation → Fixed parameter tractability

Keywords and phrases edge clique partition, edge clique cover, independence number, parameterized complexity, above guarantee

Digital Object Identifier 10.4230/LIPIcs.ESA.2025.43

Related Version *Full Version*: `https://arxiv.org/abs/2506.21216` [21]

Funding *Fedor V. Fomin*: Supported by the Research Council of Norway under BWCA project (grant no. 314528).

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

1 Introduction

Covering and partitioning the edges of a graph into cliques are fundamental combinatorial problems that lie at the intersection of combinatorial optimization, graph theory, and complexity theory. In the Edge Clique Cover (ECC) problem one seeks to cover all edges of a graph with as few cliques as possible, while in Edge Clique Partition (ECP) the same objective is pursued under the additional constraint that each edge must lie in exactly one of the chosen cliques. Both problems were extensively studied from various algorithmic perspectives (See Section 1.2 for an overview), in particular from parameterized algorithms.

ECC and ECP are known to be fixed-parameter tractable (FPT) when parameterized by the number of cliques in the cover or partition [24, 40]. While this parameterization is meaningful for dense graphs, it becomes less relevant for sparse graphs (e.g., graphs with bounded maximum degree or planar graphs), where the size of a clique cover or partition usually scales proportionally with the number of vertices. In such cases, a brute-force algorithm that guesses all possible partitions already runs in FPT time under this parameter choice.

These observations naturally lead us to explore a different parameterization – one that captures a lower bound on the size of the cover or partition. In particular, for any graph G without isolated vertices, an edge clique cover or partition must contain at least $\alpha(G)$ cliques, where $\alpha(G)$ denotes the size of a maximum independent set of G. Empirical studies of ECC (see, e.g., [29, Table 3]) indicate that in many real-world instances, after suitable preprocessing, the number of cliques in the cover is *very* close to $\alpha(G)$. This observation motivates us to investigate ECC and ECP in the setting of parameterization *above* α. Concretely, we initiate the study of the following parameterized problems.

Edge Clique Partition Above Independent Set (ECP/α) parameterized by k

Input:	A graph G without isolated vertices, an integer $k \geq 0$.
Task:	Find an edge clique partition of G with at most $\alpha(G) + k$ cliques.

Edge Clique Cover Above Independent Set (ECC/α) parameterized by k

Input:	A graph G without isolated vertices, an integer $k \geq 0$.
Task:	Find an edge clique cover of G with at most $\alpha(G) + k$ cliques.

While ECC and ECP are both FPT when parameterized by the number of cliques, their above-guarantee variants – ECC/α and ECP/α– exhibit notably different and intriguing behaviors.

1.1 Our Results

Our first result establishes parameterized complexity of Edge Clique Partition Above Independent Set (ECP/α). It is worth to note that the running time of our algorithm matches the best known running time of the algorithm for ECP [18]. Moreover, as we show, any improvement in the running time for ECP would improve the running time in our theorem too.

▶ **Theorem 1.** *ECP/α can be solved in $2^{\mathcal{O}(k^{3/2}\log k)} \cdot n^{\mathcal{O}(1)}$ time.*

We underline that the input of ECP/α (or ECC/α) does not contain the value of $\alpha(G)$. Interestingly, while computing the maximum independent set is well-known to be intractable, we show in Lemma 12 that, given an instance (G, k) of ECP/α, in $2^{\mathcal{O}(k^{3/2}\log k)} \cdot n^{\mathcal{O}(1)}$ time we can either compute $\alpha(G)$ or conclude that (G, k) is a no-instance. This algorithm is used as a subroutine in Theorem 1 and, in fact, our algorithm for ECP/α either outputs an edge clique cover of size at most $\alpha(G) + k$ together with $\alpha(G)$ or correctly reports a no-instance.

While both ECP and ECC are FPT when parameterized by the number of cliques in the solution, their above-guarantee variants exhibit drastically different behaviors. The following theorem shows that for every fixed integer $k \geq 2$, deciding whether a graph can be covered by at most $\alpha(G) + k$ cliques is NP-complete even on prefect graphs for which the independence number can be computed in polynomial time [26]. Consequently, this places ECC/α in the class of Para-NP-complete problems.

▶ **Theorem 2.** *For every $k \geq 2$, ECC/α is NP-complete. Furthermore, the hardness holds even on perfect graphs.*

The condition in Theorem 2 concerning k is tight: for $k < 2$, the problem is solvable in polynomial time. Taken together, Theorems 2 and 3 establish a dichotomy result on the complexity of ECC/α for all values of k.

▶ **Theorem 3.** *ECC/α admits polynomial-time algorithms for $k \in \{0, 1\}$.*

Despite the intractability of ECC/α on general graphs, for certain classes of sparse graphs it is possible to design FPT algorithms. We summarize these findings in the following theorem. Here and further, the statements whose proof are omitted in this extended abstract are labeled ($\star$). The proofs can be found in the full version of this paper [21].

▶ **Theorem 4 ($\star$).** *ECC/α admits parameterized algorithms with the following running times:*
- $4^{\binom{\omega}{2} \cdot k} \cdot n^{\mathcal{O}(1)}$ *on graphs with clique number ω,*
- $2.081^{(d-1)k} \cdot n^{\mathcal{O}(1)}$ *on graphs of degeneracy $d \geq 3$,*
- $1.619^{k} \cdot n^{\mathcal{O}(1)}$ *on 2-degenerate graphs,*
- $f(H)^{\sqrt{k}} \cdot n^{\mathcal{O}(1)}$ *on H-minor-free graphs.*

Neither of ω, d, H should be given to the corresponding algorithm explicitly.

Complementing this result, we observe that, under the Exponential Time Hypothesis, the dependency on k cannot be improved in neither of the algorithms; additionally, the dependency on d is almost optimal. We refer the reader to the full version of this paper [21] for proper formal discussion.

1.2 Related Work

ECC and ECP are classical combinatorial problems [16, 28, 37, 46], whose origins can be traced back to fundamental questions posed by Boole in 1868 [7]. Over time, these problems have been studied under various names, including Covering by Cliques (Problem GT17) and Intersection Graph Basis (Problem GT59) in Garey and Johnson's compendium [23], as well as Keyword Conflict [34].

From a practical standpoint, ECC and ECP arise in diverse application areas such as computational geometry [2], applied statistics [25, 43], networks [27], compiler optimization [44], or bioinformatics [19]. Due to their broad importance, both ECC and ECP have been investigated from multiple perspectives, including approximation algorithms and inapproximability bounds [3, 38], heuristics [4, 25, 34, 36, 43, 44], and polynomial-time solutions for special graph classes [8, 23, 30, 31, 32, 39, 42].

Moreover, the relation between the smallest edge clique cover and the maximum independent set plays another important role in practical applications. On large instances, it may be infeasible to compute the optimal edge clique cover, and only heuristical methods could be applied within a reasonable timeframe. Finding a suffiently large independent set serves then as a certificate for the quality of the solution, in terms of the size of the edge clique cover: If, say, there is only a 1% difference between the found independent set and clique cover, then the smallest clique cover is also bound to lie in this small interval. As observed by Hevia et al. [29], this phenomenon occurs frequently on real-world instances. Simultaneously, this certifies that the difference between the smallest edge clique cover and the largest independent set is also small on these instances.

Both ECC and ECP are known to be NP-complete even in restricted graph classes [42, 9, 30]. In particular, ECC remains NP-complete and even when the input graph is planar [9] or has bounded degree [30]. Furthermore, Lund and Yannakakis [38] showed that ECC is not approximable within a factor of n^{ε} (for some $\varepsilon > 0$) unless P=NP.

In particular, a significant amount of work in the algorithms engineering community is devoted to the studies of ECC on sparse graphs, whose independence number is large. For example, Abdullah and Hossain [1], and Conte, Grossi, and Marino [11] studied ECC on d-degenerate graphs. Blanchette, Kim, and Vetta in [6] obtained an PTAS for ECC on planar graphs as well as an FPT algorithm parameterized by the treewidth of a graph.

Within the realm of parameterized complexity, a natural parameterization is by the number of cliques k. Gramm et al. [24] pioneered this direction by presenting simple reduction rules that produce a kernel of size at most 2^k, making it one of the earliest examples of kernelization techniques described in textbooks [12, 22, 41]. Later, Cygan, Pilipczuk, and Pilipczuk [13] showed that there is no algorithm solving ECC in time $2^{2^{o(k)}} \cdot n^{\mathcal{O}(1)}$ unless the Exponential Time Hypothesis (ETH) fails.

Mujuni and Rosamond [40] established that ECP is FPT when parameterized by k, the number of cliques in the solution. Feldmann, Issac, and Rai [18] improved the running time to $2^{k^{3/2} \log k} \cdot n^{\mathcal{O}(1)}$. Fleischer and Wu [20] devised an algorithm for planar graphs running in $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$, and for graphs of degeneracy d they achieved $2^{dk} \cdot k \cdot n^{\mathcal{O}(1)}$ running time.

1.3 Organization of the Paper

In Section 2, we introduce basic notions and provide auxiliary results. Section 3 contains the proof of Theorem 1. In Section 4, we show the dichotomy for ECC/α on general graphs, proving Theorems 2 and 3. Finally, in Section 5, we give algorithmic results and establish computational lower bounds for ECC/α on sparse graph classes, as stated in Theorem 4. Due to the space restrictions, we omit proofs of some of the results. Several parts of Section 5 are also omitted. The omitted proofs and parts can be found in the full version of this paper [21].

2 Definitions and Preliminaries

Graphs. In this paper, we consider simple undirected graphs and refer to the textbook by Diestel [15] for notions that are not defined here. We use $V(G)$ and $E(G)$ to denote the set of vertices and the set of edges of G, respectively. We use n and m to denote the number of vertices and edges in G, respectively, unless this creates confusion. For a vertex subset $X \subseteq V(G)$, we use $G[X]$ to denote the subgraph of G induced by the vertices of X and $G - X$ to denote $G[V(G) \setminus X]$. For a vertex $v \in V(G)$, we write $N_G(v) = \{u \in V(G) \mid uv \in E(G)\}$ to denote the *open neighborhood* of v, and we use $N_G[v] = N_G(v) \cup \{v\}$ for the *closed neighborhood*. For $X \subseteq V(G)$, $N_G(X) = \left(\bigcup_{v \in X} N_G(v)\right) \setminus X$ and $N_G[X] = \bigcup_{v \in X} N_G[v]$. We denote by $d_G(v) = |N_G(v)|$ the *degree* of a vertex v; a vertex of degree zero is *isolated*. The minimum degree is denoted by $\delta(G)$. A graph G is *d-degenerate* for an integer $d \geq 0$ if

$\delta(H) \leq d$ for any subgraph H of G. By $\mathrm{dg}(G)$ we denote the degeneracy of G, i.e. minimum d for which G is d-degenerate. A graph H is a *minor* of G if the graph isomorphic to H can be obtained from G by vertex/edge deletions and edge contractions. A graph G is *H-minor-free* if H is not a minor of G. A set of pairwise adjacent vertices is called a *clique*, and a set of pairwise non-adjacent vertices is *independent*. The maximum size of an independent set in G is denoted by $\alpha(G)$, and the maximum size of a clique in G is denoted by $\omega(G)$. A vertex $v \in V(G)$ is said to be *simplicial* if $N_G[v]$ is a clique, and we say that a clique K is *simplicial* if K equals $N_G[v]$ for a simplicial vertex v. An edge $uv \in E(G)$ is *covered* by a clique K if $u, v \in K$. A set $\mathcal{K}$ of cliques is a *vertex clique cover* if for every $v \in V(G)$, there is $K \in \mathcal{K}$ such that $v \in K$. We say that $\mathcal{K}$ is an *edge clique cover* if for every $uv \in E(G)$, there is $K \in \mathcal{K}$ with $u, v \in K$, and $\mathcal{K}$ is an *edge clique partition* if $\mathcal{K}$ is an edge clique cover such that any two cliques in $\mathcal{K}$ has at most one common vertex. We use $\mathrm{ecc}(G)$ and $\mathrm{ecp}(G)$ to denote the minimum size of an edge clique cover and partition of G, respectively.

Parameterized Complexity. We refer to the textbook by Cygan et al. [12] for an introduction to the area. We remind that a *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$ where Σ^* is a set of strings over a finite alphabet Σ. An input of a parameterized problem is a pair (x, k) where x is a string over Σ and $k \in \mathbb{N}$ is a *parameter*. A parameterized problem is *fixed-parameter tractable* (or FPT) if it can be solved in $f(k)\cdot|x|^{\mathcal{O}(1)}$ time for some computable function f. The complexity class FPT contains all fixed-parameter tractable parameterized problems. We use the *Exponential Time Hypothesis* (ETH) of Impagliazzo, Paturi, and Zane [33] to obtain conditional computational lower bounds. Under ETH, there is a positive real δ such that 3-Satisfiability (3-SAT) with n variables and m clauses cannot be solved in $2^{\delta n} \cdot (n+m)^{\mathcal{O}(1)}$ time; in particular, 3-SAT cannon be solved in time subexponential in n.

We need the following auxiliary results about edge clique covers and independent sets. First, we observe that $\alpha(G)$ gives a lower bound for $\mathrm{ecc}(G)$ and $\mathrm{ecp}(G)$.

▶ **Observation 5** ($\star$). *For a graph G without isolated vertices $\alpha(G) \leq \mathrm{ecc}(G) \leq \mathrm{ecp}(G)$.*

Notice that the absence of isolated vertices is crucial for the lower bound of $\mathrm{ecc}(G)$ and $\mathrm{ecp}(G)$. We also remark that that excluding isolated vertices is essential for our algorithmic results for ECP$/\alpha$ and ECP$/\alpha$.

▶ **Observation 6** ($\star$). *For any graph class $\mathcal{G}$ closed under adding pendent neighbors and isolated vertices such that ECP (ECC) is* NP*-complete on $\mathcal{G}$, it is* NP*-complete to decide whether a graph $G \in \mathcal{G}$ given together with its maximum independent set has an edge clique partition (cover) of size at most $\alpha(G)$.*

The following lemma plays the most fundamental role for ECC$/\alpha$ and ECP$/\alpha$.

▶ **Lemma 7.** *Let G be a graph without isolated vertices, and let $\mathcal{C}$ be an edge clique cover (partition) of G of size at most $\alpha(G) + k$ for an integer $k \geq 0$. Then*

(i) *at most k vertices of any maximum independent set are non-simplicial,*

(ii) *at most $2k$ cliques of $\mathcal{C}$ are non-simplicial.*

Proof. As an edge clique partition is an edge clique cover, it is sufficient to show the claim when $\mathcal{C}$ is an edge clique cover of size at most $\alpha(G)+k$. Let X be a maximum independent set of G. Clearly, any clique of G contains at most one vertex of X by independence. Because G has no isolated vertices, each vertex $v \in X$ is included in at least one clique of $\mathcal{C}$. Furthermore, v is simplicial if v is included in a single clique of $\mathcal{C}$. Since $|\mathcal{C}| \leq \alpha(G) + k$, we obtain that at least $\alpha(G) - k$ vertices of X are contained in exactly one clique of $\mathcal{C}$. Therefore, at least

$\alpha(G) - k$ vertices of X are simplicial, and at least $\alpha(G) - k$ cliques of $\mathcal{C}$ are simplicial. This means that at most k vertices of X are non-simplicial and proves (i). To show (ii), note that because at least $\alpha(G) - k$ cliques of $\mathcal{C}$ are simplicial, at most $2k$ cliques could be non-simplicial. This concludes the proof. ◀

In particular, this lemma implies that ECC/α and ECP/α are trivial for $k = 0$.

▶ **Observation 8** ($\star$). *ECC/α and ECP/α can be solved in polynomial time for $k = 0$.*

We use Lemma 7 to argue that for graphs admitting an edge clique cover of bounded size, we can compute $\alpha(G)$.

▶ **Lemma 9** ($\star$). *Let G be a graph and let $\mathcal{C}$ be an edge clique cover (partition) of G of size at most k. Then $\alpha(G)$ can be computed in $\mathcal{O}(2^k \cdot k^2 n)$ time.*

We remark that our algorithm from Lemma 9 can be easily modified to output an independent set of maximum size.

3 Clique Partition above Independent Set

In this section, we prove Theorem 1. For this, we need several auxiliary results. We start with the classical result of de Bruijn and Erdős from 1948 [14].

▶ **Proposition 10** (de Brujin and Erdős [14]). *For any $n \geq 3$, an edge clique partition of K_n into cliques of size at most $n - 1$ contains at least n cliques.*

To solve ECP/α, we use as a black box an algorithm for ECP parameterized by the number of cliques in a solution. The state-of-the-art FPT algorithm was given by Feldmann, Issac, and Rai [18].

▶ **Proposition 11** (Feldmann, Issac, Rai [18]). *ECP can be solved in $2^{\mathcal{O}(k^{3/2} \log k)} \cdot n^{\mathcal{O}(1)}$ time where k is the maximum number of cliques in a solution. Furthermore, the algorithm outputs an edge clique partition of size at most k if it exists.*

Next, we show that combining Lemma 7, Lemma 9, and Proposition 11, we can compute $\alpha(G)$ for graphs admitting an edge clique partition of size at most $\alpha(G) + k$.

▶ **Lemma 12.** *There is an algorithm with running time $2^{\mathcal{O}(k^{3/2} \log k)} \cdot n^{\mathcal{O}(1)}$ that, given an instance (G, k) of ECP/α, either computes $\alpha(G)$ or correctly reports that (G, k) is a no-instance of ECP/α.*

Proof. Let G be a graph without isolated vertices. We compute the set $\mathcal{S}$ of all simplicial cliques and select a simplicial vertex in each simplicial clique. Denote by X the obtained set of simplicial vertices. It is well-known that G has a maximum independent set U such that $X \subseteq U$. This implies that $\alpha(G) = \alpha(H) + |\mathcal{S}|$ where $H = G[W]$ for $W = V(G) \setminus \bigcup_{S \in \mathcal{S}} S$. If G admits an edge clique partition of size at most $\alpha(G) + k$ then, by Lemma 7, at most $2k$ cliques of the partition are non-simplicial. Notice that the edges of H can be covered only by non-simplicial cliques. Therefore, H should have an edge clique partition of size at most $2k$. We apply the algorithm from Proposition 11 to H and check in $2^{\mathcal{O}(k^{3/2} \log k)} \cdot n^{\mathcal{O}(1)}$ time whether H has an edge clique partition of size at most $k' = 2k$. If the algorithm reports that such a partition does not exist then we conclude that (G, k) is a no-instance of ECP/α. Otherwise, the algorithm outputs an edge clique partition $\mathcal{C}$ of size at most $2k$. Given this partition, we use the algorithm from Lemma 9 to compute $\alpha(H)$ in $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time. Then we output $\alpha(G) = \alpha(H) + |\mathcal{S}|$. Because simplicial cliques can be found in polynomial time, for example, by the algorithm of Kloks, Kratsch, and Müller [35], the overall running time is $2^{\mathcal{O}(k^{3/2}\log k)} \cdot n^{\mathcal{O}(1)}$. This concludes the proof. ◀

By the following claim, which is proved by making use of Proposition 10, we show that all cliques of size at least $6k+1$ that belong to any solution to an instance of ECP/α can be found in polynomial time.

▶ **Lemma 13** ($\star$). *There is a polynomial algorithm that, given an instance (G, k) of ECP/α for $k \geq 1$, either outputs a set $\mathcal{K}$ of cliques of G such that*

(i) *each clique $C \in \mathcal{K}$ is included in any edge clique partition of G of size at most $\alpha(G)+k$,*
(ii) *for every edge clique partition $\mathcal{C}$ of G of size at most $\alpha(G)+k$ and any clique $C \in \mathcal{C}$ of size at least $6k+1$, $C \in \mathcal{K}$,*
(iii) *for any two distinct cliques $C_1, C_2 \in \mathcal{K}$, $|C_1 \cap C_2| \leq 1$,*

or correctly concludes that (G, k) is a no-instance.

Now we are ready to show Theorem 1 which we restate.

▶ **Theorem 1.** *ECP/α can be solved in $2^{\mathcal{O}(k^{3/2}\log k)} \cdot n^{\mathcal{O}(1)}$ time.*

Proof. Let (G, k) be an instance of ECP/α. As pointed out in Observation 8, the problem is trivial if $k = 0$. Hence, we assume that $k \geq 1$. First, we call the algorithm from Lemma 12 for (G, k). The algorithm either computes $\alpha(G)$ or correctly reposts that (G, k) is a no-instance of ECP/α. If we obtain a no-instance, we report it and stop. We assume from now on that this is not the case, and the value $\alpha(G)$ is known to us.

In the following step, we construct a partial solution $\mathcal{K}$ using the algorithm from Lemma 13. This algorithm works in polynomial time and either outputs a set $\mathcal{K}$ of cliques of G such that

(i) each clique $C \in \mathcal{K}$ is included in any edge clique partition of G of size at most $\alpha(G)+k$,
(ii) for every edge clique partition $\mathcal{C}$ of G of size at most $\alpha(G)+k$ and any clique $C \in \mathcal{C}$ of size at least $6k+1$, $C \in \mathcal{K}$,
(iii) for any two distinct cliques $C_1, C_2 \in \mathcal{K}$, $|C_1 \cap C_2| \leq 1$,

or correctly concludes that (G, k) is a no-instance. If the algorithm returns that (G, k) is a no-instance then we return the answer and stop. We also conclude that (G, k) is a no-instance if $|\mathcal{K}| > \alpha(G) + k$. From now on, we assume that $\mathcal{K}$ is a set of cliques that are included in any edge clique partition of size at most $\alpha(G) + k$ such that any two distinct cliques of $\mathcal{K}$ do not cover the same edge.

Next, we list all distinct simplicial cliques of G and denote by $\mathcal{S}$ the set of simplicial cliques. We say that a clique $S \in \mathcal{S}$ is *broken* in a (potential) solution to (G, k), if S is not included in the edge clique partition. Notice that if S is a broken simplicial clique, then at least two distinct non-simplicial cliques should be used to cover the edges incident to the corresponding simplicial vertex. By Lemma 7, any edge clique partition of size at most $\alpha(G) + k$ can include at most $2k$ non-simplicial cliques. Thus, for any solution, the number of broken simplicial cliques should not exceed k. Our algorithm identifies the set $\mathcal{B}$ of broken cliques in a solution.

We initialize $\mathcal{B}$ using the following branching algorithm. Consider the auxiliary graph H whose nodes are simplicial cliques. Two distinct nodes $S_1, S_2 \in \mathcal{S}$ are adjacent in H if and only if $|S_1 \cap S_2| \geq 2$. Observe that if $|S_1 \cap S_2| \geq 2$ then either S_1 or S_2 should be broken in a solution. Thus, the set of broken cliques in any solution should be a vertex cover of H of size at most k. We can decide in $\mathcal{O}(2^k \cdot n)$ time whether H has a vertex cover of size at most k and, moreover, we can list all inclusion minimal vertex covers using the standard recursive branching algorithm for VERTEX COVER (see [12]). If H has no vertex cover of size at most k, we report that (G,k) is a no-instance of ECP$/\alpha$ and stop. Assume that this is not the case. Then we obtain the list $\mathcal{L}$ of all inclusion minimal vertex covers of H of size at most k. Notice that $|\mathcal{L}| \leq 2^k$. Then if (G,k) admits an edge clique partition of size at most $\alpha(G)+k$, there is $L \in \mathcal{L}$ such that $L \subseteq \mathcal{B}$ – the set of broken cliques in $\mathcal{C}$. To initialize $\mathcal{B}$, we branch on all possible choices of $L \in \mathcal{L}$ and set $\mathcal{B} := L$. If we find a solution for some choice of L, we output it and stop. Otherwise, if we fail to find an edge clique partition of size at most $\alpha(G)+k$ for any initial selection of $\mathcal{B}$, we report that (G,k) is a no-instance of ECP$/\alpha$.

From now on, we assume that the initial choice of $\mathcal{B}$ is fixed from one of at most 2^k choices given by $\mathcal{L}$. We update $\mathcal{B}$ by adding cliques that get broken because of the cliques in the partial solution. Notice that if $S \notin \mathcal{K}$ is a simplicial clique covering the same edge as one of the cliques in the partial solution, then S has to be broken. This implies the correctness of the following step:

- For every simplicial clique $S \notin \mathcal{B} \cup \mathcal{K}$ such that $|S \cap K| \geq 2$ for some $K \in \mathcal{K}$, set $\mathcal{B} := \mathcal{B} \cup \{S\}$.
- If $|\mathcal{B}| > k+1$ then stop and discard the current choice of L.

Assume that the algorithm does not stop in this step. We call the set $\mathcal{B}$ obtained in this step a *base set of broken cliques*.

We say that a simplicial clique S is *free* if $S \notin \mathcal{B} \cup \mathcal{K}$, and we use $\mathcal{F}$ to denote the set of free cliques. Lemma 13 and the construction of $\mathcal{B}$ imply the following property.

▷ Claim 14. For any two distinct cliques $S_1, S_2 \in \mathcal{F} \cup \mathcal{K}$, $|S_1 \cap S_2| \leq 1$.

This means that no edge of G is covered by two cliques of $\mathcal{F} \cup \mathcal{K}$. In the final step of our algorithm, we try to extend $\mathcal{F} \cup \mathcal{K}$ to an edge clique partition of G of size at most $\alpha(G)+k$. If we fail, then we recurse by including one of the free cliques to the set of broken cliques. Formally, we use the subroutine Extend$(\mathcal{F},\mathcal{B})$ in Algorithm 1 which either finds a solution extending $\mathcal{K}$, such that all cliques in $\mathcal{B}$ are broken, or discards the current choice of $\mathcal{F}$ and $\mathcal{B}$.

The description of the algorithm is finished. To argue correctness, observe that in each recursive call of the algorithm, we increase the size of $\mathcal{B}$ and stop if $|\mathcal{B}| > k$ in line 2. This means that Extend$(\mathcal{F},\mathcal{B})$ is finite. If this subroutine outputs a set of cliques $\mathcal{C}^*$ then it is a valid edge partition of G. First, because of Claim 14, no two cliques intersect. Second, the cover $\mathcal{C}$, constructed by the algorithm from Proposition 11 in line 5, outputs an edge clique partition for the edges which are not covered by the cliques of $\mathcal{F} \cup \mathcal{K}$. Since we verify that $|\mathcal{C}^*| \leq \alpha(G)+k$ in line 7, if the algorithm outputs some $\mathcal{C}^*$ then it is a valid solution to (G,k). Thus, we have to show that if (G,k) is a yes-instance of ECP$/\alpha$, then the algorithm necessarily finds a solution. For this, we show the following claim.

▷ Claim 15 (⋆). Suppose that G has an edge clique partition of size at most $\alpha(G)+k$ such that all simplicial cliques in $\mathcal{B}$ are broken. Suppose also that $\mathcal{B}$ contains the base set of broken cliques, and let $\mathcal{F}$ be the set of simplicial cliques S such that $S \notin \mathcal{B} \cup \mathcal{K}$. Then Extend$(\mathcal{F},\mathcal{B})$ outputs a solution to (G,k).

To complete the correctness proof, observe that if (G,k) is a yes-instance of ECP$/\alpha$ then there is $L \in \mathcal{L}$ such that every clique of L is broken in a solution. Then for the base set of broken cliques $\mathcal{B}$ constructed for L, each clique of $\mathcal{B}$ should be broken. By Claim 15, Extend$(\mathcal{F},\mathcal{B})$ called for this $\mathcal{B}$ and the corresponding set of free cliques $\mathcal{F}$ should output an edge clique partition of size at most $\alpha(G)+k$. This completes the correctness proof.

We conclude the proof with a claim explaining the running time bound.

Algorithm 1 Extend($\mathcal{F}, \mathcal{B}$).

Input: $\mathcal{F}$, $\mathcal{B}$
Result: An edge clique cover of G or No
begin
  if $|\mathcal{B}| > k$ **then return** No and quit;
  construct the graph Q with the edge set $R = \{uv \in E(G) \mid \text{there is no clique } K \in \mathcal{F} \cup \mathcal{K} \text{ s.t. } u, v \in K\}$, whose vertex set is the set of the endpoints of the edges of R; note that R is the set of edges which are not covered by $\mathcal{F} \cup \mathcal{K}$;
  if $|V(Q)| > 12k^2$ **then return** No and quit;
  use the algorithm from Proposition 11 to find an edge clique partition $\mathcal{C}$ of Q of minimum size $k' \leq 2k$ if such a partition exists;
  if *the algorithm returns* $\mathcal{C}$ **then**
    if $|\mathcal{C}^*| \leq \alpha(G) + k$ *for* $\mathcal{C}^* = \mathcal{C} \cup \mathcal{F} \cup \mathcal{K}$ **then return** $\mathcal{C}^*$ and quit;
  end
  foreach $uv \in E(G)$ *such that* $u, v \in V(Q)$ *and there is* $F \in \mathcal{F}$ *with* $u, v \in F$ **do**
    if *a call to* Extend($\mathcal{F} \setminus \{F\}, \mathcal{B} \cup \{F\}$) *returns a solution* $\mathcal{C}^*$ **then**
      return $\mathcal{C}^*$ and quit;
    end
  end
  return No
end

▷ Claim 16 (⋆). The algorithm runs in $2^{\mathcal{O}(k^{3/2} \log k)} \cdot n^{\mathcal{O}(1)}$ time.

The proof is complete. ◀

In the conclusion of this section, we remark that the algorithm of Proposition 11 is a bottleneck for the running time of our algorithm for ECP/α. More precisely, given an algorithm solving ECP in time $f(k) \cdot n^{\mathcal{O}(1)}$, the running time of our algorithm is $k^{\mathcal{O}(k)} f(2k) \cdot n^{\mathcal{O}(1)}$. Furthermore, consider the graph G' obtained from a graph G by adding a pendent neighbor to each vertex of G. Since $\operatorname{ecp}(G') = \alpha(G') + \operatorname{ecp}(G)$, a faster algorithm for ECP/α would improve the algorithm of Proposition 11.

4 Clique Cover above Independent Set

In this section, we establish the dichotomy result for ECC/α by showing Theorem 2 and Theorem 3. For this, we need some auxiliary results about relations of Edge Clique Cover Above Independent Set with other problems which will be useful also in Section 5. Therefore, we put them in a separate subsection.

4.1 Reductions between ECC/α, Annotated ECC, and k-Coloring

Following Orlin [42], we consider the generalization of ECC where the aim is to cover a subset of edges.

ANNOTATED EDGE CLIQUE COVER (ANNOTATED ECC) parameterized by k

Input:	A graph G, an edge set $B \subseteq E(G)$, an integer k.
Task:	Find k cliques in G, such that each edge in B belongs to at least one clique.

Recall that $\mathrm{ecc}(G)$ denotes the size of the smallest edge clique cover of G. Similarly, in the context of ANNOTATED ECC, for a graph G and a subset of its edges $B \subseteq E(G)$, we denote by $\mathrm{aecc}_B(G)$ the size of the smallest collection of cliques in G that cover all edges in B.

▶ **Lemma 17** ($\star$). *ECC/α admits an* FPT*-Turing-reduction to* ANNOTATED ECC*, where an instance (G, k) is reduced to at most $2k$ instances (G', R', k') such that G' is an induced subgraph of G and $k' \leq 2k$. The reduction works in time $4^k \cdot n^{\mathcal{O}(1)}$.*

The running time of Lemma 17 is dominated by computing the maximum independent set in an induced subgraph, and the reduction can be performed more efficiently on certain graph classes, where instead of Lemma 9 one can compute the independent set directly with a faster algorithm. To accomodate for this, we state the following corollary of Lemma 17.

▶ **Corollary 18** ($\star$). *Let $\mathcal{G}$ be a hereditary graph class. ECC/α admits an* FPT*-reduction to* ANNOTATED ECC*, where an instance (G, k) with $G \in \mathcal{G}$ is reduced to at most $2k$ instances (G', R', k') such that G' is an induced subgraph of G and $k' \leq 2k$. If it can be determined in time $f(k) \cdot n^{\mathcal{O}(1)}$ whether a graph in $\mathcal{G}$ has an independent set of size k, then the reduction admits the same running time bound.*

Next, we show that ANNOTATED ECC can be, in turn, reduced to the k-COLORING problem. We note that the reduction is almost the same as the standard conversion from ECC to VERTEX CLIQUE COVER, as introduced by Kou, Stockmeyer and Wong [36].

▶ **Lemma 19** ($\star$). *ANNOTATED ECC is polynomial-time reducible to k-COLORING on a graph with $|B|$ vertices.*

As k-COLORING is polynomial-time solvable when $k \leq 2$, we immediately get the following corollary.

▶ **Corollary 20.** *ANNOTATED ECC is polynomial-time solvable for $k \leq 2$.*

We note that a polynomial-time reduction from k-COLORING or VERTEX CLIQUE COVER to ANNOTATED ECC follows immediately from a reduction to ECC.

4.2 Dichotomy for ECC/α

In this subsection, we prove that ECC/α is Para-NP-complete when parameterized by k. More precisely, we show that the problem is NP-complete for every $k \geq 2$ on instances where the graph is perfect without isolated vertices. We complement this result by proving that for $k = 0$ or 1, the problem is in P. We start with obtaining the following result for ANNOTATED ECC. The arguments are similar to the arguments of Orlin [42].

▶ **Lemma 21** ($\star$). *ANNOTATED ECC is* NP*-complete on co-bipartite graphs for every $k \geq 3$. Furthermore, the hardness holds for the case when B is a perfect matching between the bipartition sets.*

We use Lemma 21 to prove Theorem 2. We restate it here.

▶ **Theorem 2.** *For every $k \geq 2$, ECC/α is* NP*-complete. Furthermore, the hardness holds even on perfect graphs.*

Proof. We use Lemma 21 and reduce from Annotated ECC.

Let (G, B, k) be an instance of Annotated ECC where G is an n-vertex co-bipartite graph for $n \geq 2$. We also assume that B is a perfect matching in G, that is, each vertex of G is incident to an edge of B. We set $R = E(G) \setminus B$ and construct the graph G' as follows.

- Construct a copy of G.
- Construct a vertex v and make it adjacent to each vertex of G.
- For each vertex $x \in V(G)$, construct a vertex u_x and make it adjacent to x.
- For every edge $e = xy \in R$, construct a vertex w_e and make it adjacent to x and y.

Because G is a co-bipartite graph with at least two vertices that has a perfect matching, G has no isolated vertices. Then the construction implies that the same holds for G'. Notice that G' is a perfect graph. To see this, we use the perfect graph theorem [10]. Trivially, a vertex of degree one does not belong to any odd hole or antihole. For any vertex w_e for $e \in R$, w_e does not belong to any odd hole because the neighbors of w_e are adjacent, and w_e does not belong to any antihole with at least 7 vertices because the degree of w_e is two. Since the graph obtained from the co-bipartite graph G by adding the universal vertex v is co-bipartite, this graph is perfect. This immediately implies that G' is perfect.

Consider the set $I = \{v\} \cup \{u_x \mid x \in V(G)\} \cup \{w_e \mid e \in R\}$. By the construction of G', I is an independent set of size $|R| + n + 1$. We claim that I is a maximum independent set. To see this, note that the vertices of $I' = \{u_x \mid x \in V(G)\} \cup \{w_e \mid e \in R\}$ are simplicial. Therefore, there is a maximum independent set of G' containing I'. Since v is the unique vertex nonadjacent to the vertices of I', we obtain that the size of I is maximum. Thus, $\alpha(G') = |R| + n + 1$.

We set $k' = k - 1$ and claim the following.

▷ Claim 22 (⋆). (G, B, k) is a yes-instance of Annotated ECC if and only if (G', k') is a yes-instance of ECC/α.

This concludes the proof of the theorem. ◀

As a corollary of Lemma 19 and Corollary 20, we obtain that when $k = 0$ or $k = 1$, ECC/α can be solved in polynomial time. Thus, combining Theorem 2 and Theorem 3, we get the computational complexity dichotomy for the problem with respect to k.

▶ **Theorem 3.** *ECC/α admits polynomial-time algorithms for $k \in \{0, 1\}$.*

Proof. Given an instance (G, k) of ECC/α for $k \leq 1$, the algorithm from Lemma 17 in polynomial time reduces the problem to solving at most two instances (G', B, k') of Annotated ECC where $k' \leq 2$. Then we apply the algorithm from Corollary 20 to the obtained instances and solve the problem in polynomial time. ◀

Using the result of Orlin [42] about the NP-completeness of the Edge Biclique Cover problem, it is easy to see that ECC is NP-complete on co-bipartite graphs. This result immediately implies that ECC/α is NP-complete on co-bipartite graphs. We provide the proof for completeness in [21].

▶ **Observation 23** (⋆). *ECC/α is* NP*-complete on co-bipartite graphs.*

Because ECC/α is NP-complete on co-bipartite graphs, we have that for every integer $p \geq 2$, NP-complete on graphs G with $\alpha(G) \leq p$. It is trivial to see that if $\alpha(G) = 1$ then G is a complete graph and, therefore, the edges of G can be covered by a single clique.

Theorem 2 and Observation 23 imply that ECC/α is Para-NP-hard when parameterized by either k or the independence number. Because ECC is FPT when parameterized by the number of cliques in a solution [24], it is straightforward to make the following observation.

▶ **Observation 24.** *ECC/α is* FPT *when parameterized by both k and the independence number of the input graph.*

5 Covering Sparse Graphs

In this section, we design FPT algorithms for ECC/α for several well-studied graph classes, where ECC or Independent Set remain NP-complete. We build on Lemma 17 (and Corollary 18), that allow us to transform parameterized algorithms for Annotated ECC (and Independent Set) on a graph class $\mathcal{G}$, into parameterized algorithms for ECC/α on $\mathcal{G}$. The only requirement for $\mathcal{G}$ is being closed under deletion of simplicial vertices.

In the first two subsections of this section, we give parameterized algorithms for Annotated ECC on K_r-free, d-degenerate and H-minor-free graphs. In the third subsection, we combine these algorithms into corresponding algorithms for ECC/α, proving Theorem 4. Finally, we complement these results with lower bounds based on the Exponential Time Hypothesis.

5.1 Bounded Clique Number

Annotated ECC on K_r-free graphs can alternatively be seen as Annotated ECC parameterized by $k+\omega$, where ω is the maximum clique size in G. Naturally, we cannot have more than $\binom{\omega}{2} \cdot k$ edges in B in a yes-instance of Annotated ECC, which hints that Annotated ECC should be FPT with respect to $k + \omega$. There is, however, a crucial obstacle: computing ω is NP-hard and W[1]-hard when parameterized by ω.

Fortunately, independent sets of size more than k in G will guarantee that (G, B, k) is a no-instance. This allows us to use the classical Ramsey's theorem [45] to either conclude that the instance is negative or to find ω. Formally, we use the following algorithmic result that is derived from the Ramsey number upper bound given by Erdős and Szekeres [17].

▶ **Proposition 25** (Adaptation of a proof by Erdős and Szekeres [17]). *There is a polynomial-time algorithm that, given n-vertex graph G and two integers p, q such that $n \geq \binom{p+q-2}{p-1}$, finds either a clique of size p or an independent set of size q in G.*

We are ready to prove that Annotated ECC is FPT with respect to $k + \omega$.

▶ **Lemma 26.** *Annotated ECC is fixed-parameter tractable with respect to the combined parameter $k + \omega$. The running time of the corresponding algorithm is $2^{\binom{\omega}{2}\cdot k} \cdot n^{\mathcal{O}(1)}$. The value of ω needs not to be given in the input.*

Proof. We present an algorithm that finds a solution to the given instance (G, B, k) of Annotated ECC. We assume that G, B are not empty, as otherwise the instance is trivial. If $k \leq 2$, we use the algorithm of Corollary 20 as a subroutine to solve the instance in polynomial time. We explain how the algorithm deals with the general case $k \geq 3$ in several steps below.

Triangle-free check. The algorithm first checks whether G is triangle-free, that is, $\omega > 2$. This is performed in polynomial time. If G has $\omega = 2$, then no two edges can belong to the same clique in G, so optimal cover of edges in B consists of $|B|$ cliques. Hence, if G turns out to be triangle-free, then the algorithm reports that (G, B, k) is a yes-instance if $|B| \le k$ and no-instance otherwise.

Clique or independent set. If there is a vertex $v \in V(G)$ without incident edges in B, then there exists an optimal solution where none of the cliques contain v. The algorithm removes all such vertices from G. Now any solution to (G, B, k) should contain each vertex of G in at least one of the cliques. The algorithm then finds the largest r such that $n \ge \binom{k+r-1}{k}$ and $n < \binom{k+r}{k}$. Substituting r and $k+1$ in Proposition 25, the algorithm finds either an independent set of size $k+1$ in G or a clique of size r in G in polynomial time. An independent set of size $k+1$ guarantees that (G, B, k) is a no-instance, as it is impossible to cover each vertex of G by using at most k cliques. With this outcome, our algorithm reports a no-instance and stops. In the complementary case, we have that $\omega \ge r$ and $n < \binom{k+\omega}{k}$.

Computing ω. For an integer $p \ge 3$, $\omega < p$ can be verified in time $\mathcal{O}(p^2 \cdot n^p)$. As the next step, the algorithm computes the exact value of ω by performing these checks iteratively for consecutive integer values starting from $\max\{4, r+1\}$ up to $\omega + 1$. This routine takes $n^{\omega + \mathcal{O}(1)}$ running time. To give an upper bound on the running time in terms of k and ω, we need the following claim.

▷ Claim 27 (⋆). If a, b are positive integers with $a, b \ge 3$, then $\binom{a+b}{b} \le 2^{\frac{ab}{2}}$.

By substituting $k, \omega \ge 3$ in Claim 27, we obtain

$$n^{\omega-1} < \binom{k+\omega}{\omega}^{\omega-1} < \left(2^{\frac{k\omega}{2}}\right)^{\omega-1} = 2^{\binom{\omega}{2}\cdot k}.$$

Hence, it takes time $2^{\binom{\omega}{2}\cdot k} \cdot n^{\mathcal{O}(1)}$ for the algorithm to compute the exact value of ω.

Solution via set cover. No clique in G can cover more than $\binom{\omega}{2}$ edges in B. Thus, if $|B| > \binom{\omega}{2} \cdot k$, the algorithm reports correctly that (G, B, k) is a no-instance. Otherwise, the algorithm reduces (G, B, k) to an instance $(B, \mathcal{F}, k)$ of Set Cover. In this instance, B is treated as the universe set. Subset family $\mathcal{F}$ consists of subsets of B, and a subset $B' \subset B$ belongs to $\mathcal{F}$ if and only if all edges of B' belong simultaneously to the same clique in G. Clearly, (G, B, k) is a yes-instance if and only if B can be covered by at most k sets from $\mathcal{F}$, i.e. $(B, \mathcal{F}, k)$ is a yes-instance of Set Cover.

The Set Cover instance is constructed in time $2^{|B|} \cdot n^{\mathcal{O}(1)}$ by the algorithm. Using the celebrated subset convolution algorithm for Set Cover due to Björklund, Husfeldt and Koivisto [5] as a subroutine, our algorithm finally decides $(B, \mathcal{F}, k)$ within the same $2^{|B|} \cdot n^{\mathcal{O}(1)}$ running time bound. The overall running time is therefore dominated by $2^{\binom{\omega}{2}\cdot k} \cdot n^{\mathcal{O}(1)}$. ◀

Consequent parts of Section 5 are omitted and can be found in the full version of the paper [21].

References

1 Wali Mohammad Abdullah and Shahadat Hossain. A sparse matrix approach for covering large complex networks by cliques. In *International Conference on Computational Science*, pages 505–517. Springer, 2022. doi:10.1007/978-3-031-08757-8_43.

2 Pankaj K. Agarwal, Noga Alon, Boris Aronov, and Subhash Suri. Can visibility graphs be represented compactly? *Discrete & Computational Geometry*, 12:347–365, 1994. doi:10.1007/BF02574385.

3 Giorgio Ausiello, Pierluigi Crescenzi, Giorgio Gambosi, Viggo Kann, Alberto Marchetti-Spaccamela, and Marco Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer, 1999. doi:10.1007/978-3-642-58412-1.

4 Michael Behrisch and Anusch Taraz. Efficiently covering complex networks with cliques of similar vertices. *Theoretical Computer Science*, 355(1):37–47, 2006. doi:10.1016/j.tcs.2005.12.005.

5 Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set partitioning via inclusion-exclusion. *SIAM Journal on Computing*, 39(2):546–563, January 2009. doi:10.1137/070683933.

6 Mathieu Blanchette, Ethan Kim, and Adrian Vetta. Clique cover on sparse networks. In *Proceedings of the 14th Meeting on Algorithm Engineering & Experiments (ALENEX)*, pages 93–102. SIAM / Omnipress, 2012. doi:10.1137/1.9781611972924.10.

7 George Boole. *Of Propositions Numerically Definite*. Watts, London, 1952. Originally published in 1868, Chapter IV.

8 Márcia R. Cerioli, Luérbio Faria, Talita O. Ferreira, Carlos A. J. Martinhon, Fábio Protti, and Bruce A. Reed. Partition into cliques for cubic graphs: Planar case, complexity and approximation. *Discret. Appl. Math.*, 156(12):2270–2278, 2008. doi:10.1016/J.DAM.2007.10.015.

9 Maw-Shang Chang and Haiko Müller. On the tree-degree of graphs. In *In Proceedings of Graph-Theoretic Concepts in Computer Science, 27th International Workshop (WG 2001)*, volume 2204 of *Lecture Notes in Computer Science*, pages 44–54. Springer, 2001. doi:10.1007/3-540-45477-2_6.

10 Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Ann. of Math. (2)*, 164(1):51–229, 2006. doi:10.4007/annals.2006.164.51.

11 Alessio Conte, Roberto Grossi, and Andrea Marino. Large-scale clique cover of real-world networks. *Information and Computation*, 270:104464, 2020. doi:10.1016/J.IC.2019.104464.

12 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

13 Marek Cygan, Marcin Pilipczuk, and Michal Pilipczuk. Known algorithms for edge clique cover are probably optimal. *SIAM J. Comput.*, 45(1):67–83, 2016. doi:10.1137/130947076.

14 Nicolaas Govert de Bruijn and Pál Erdős. On a combinatorial problem. *Nederl. Akad. Wetensch., Proc.*, 51:1277–1279 = Indagationes Math. 10, 421–423, 1948.

15 Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 5th edition, 2017.

16 Pál Erdős, Adolph W. Goodman, and Lajos Pósa. The representation of a graph by set intersections. *Canadian Journal of Mathematics*, 18:106–112, 1966.

17 Pál Erdős and George Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935. URL: https://www.numdam.org/item/CM_1935__2__463_0/.

18 Andreas Emil Feldmann, Davis Issac, and Ashutosh Rai. Fixed-parameter tractability of the weighted edge clique partition problem. In *15th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 180 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 17, 16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.

19 Andres Figueroa, James Bornemann, and Tao Jiang. Clustering binary fingerprint vectors with missing values for dna array data analysis. *Journal of Computational Biology*, 11:887–901, 2004. doi:10.1089/CMB.2004.11.887.

20 Rudolf Fleischer and Xiaotian Wu. Edge clique partition of k 4-free and planar graphs. In *International Conference on Computational Geometry, Graphs and Applications*, pages 84–95. Springer, 2010. doi:10.1007/978-3-642-24983-9_9.

21 Fedor V. Fomin, Petr A. Golovach, Danil Sagunov, and Kirill Simonov. Edge clique partition and cover beyond independence, 2025. doi:10.48550/arXiv.2506.21216.

22 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization. Theory of Parameterized Preprocessing*. Cambridge University Press, 2019.

23 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W.H. Freeman, 1979.

24 Jens Gramm, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Data reduction and exact algorithms for clique cover. *ACM J. Exp. Algorithmics*, 13, 2008. doi:10.1145/1412228.1412236.

25 Jens Gramm, Jiong Guo, Falk Hüffner, Rolf Niedermeier, Hans-Peter Piepho, and Ramona Schmid. Algorithms for compact letter displays: Comparison and evaluation. *Computational Statistics & Data Analysis*, 52(2):725–736, 2007. doi:10.1016/j.csda.2006.09.035.

26 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988. doi:10.1007/978-3-642-97881-4.

27 Jean-Loup Guillaume and Matthieu Latapy. Bipartite structure of all complex networks. *Inf. Process. Lett.*, 90(5):215–221, 2004. doi:10.1016/j.ipl.2004.03.007.

28 Marshall Hall Jr. A problem in partitions. *Bulletin of the American Mathematical Society*, 47:804–807, 1941.

29 Anthony Hevia, Benjamin Kallus, Summer McClintic, Samantha Reisner, Darren Strash, and Johnathan Wilson. Solving Edge Clique Cover Exactly via Synergistic Data Reduction. In *31st Annual European Symposium on Algorithms (ESA)*, volume 274 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 61:1–61:19, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2023.61.

30 Douglas N Hoover. Complexity of graph covering problems for graphs of low degree. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 11:187–208, 1992.

31 Wen-Lian Hsu and Kuo-Hui Tsai. Linear time algorithms on circular-arc graphs. *Inf. Process. Lett.*, 40(3):123–129, 1991. doi:10.1016/0020-0190(91)90165-E.

32 Harry B. Hunt III, Madhav V. Marathe, Venkatesh Radhakrishnan, and Richard Edwin Stearns. The complexity of planar counting problems. *SIAM J. Comput.*, 27(4):1142–1167, 1998. doi:10.1137/S0097539793304601.

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

34 Eduardo Kellerman. Determination of keyword conflict. *IBM Technical Disclosure Bulletin*, 16(2):544–546, 1973.

35 Ton Kloks, Dieter Kratsch, and Haiko Müller. Finding and counting small induced subgraphs efficiently. *Inf. Process. Lett.*, 74(3-4):115–121, 2000. doi:10.1016/S0020-0190(00)00047-8.

36 Lawrence T. Kou, Larry J. Stockmeyer, and C. K. Wong. Covering edges by cliques with regard to keyword conflicts and intersection graphs. *Communications of the ACM*, 21(2):135–139, 1978. doi:10.1145/359340.359346.

37 László Lovász. On covering of graphs. In *Theory of Graphs, Proceedings of the Colloquium Tihany, Hungary, 1966*, pages 231–236. Academic Press, Budapest, 1968.

38 Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994. doi:10.1145/185675.306789.

39 Shaohan Ma, Walter D. Wallis, and Julin Wu. Clique covering of chordal graphs. *Utilitas Mathematica*, 36:151–152, 1989.

40 Egbert Mujuni and Frances A Rosamond. Parameterized complexity of the clique partition problem. In *CATS*, pages 75–78. Citeseer, 2008. URL: http://crpit.scem.westernsydney.edu.au/abstracts/CRPITV77Mujuni.html.

41 Rolf Niedermeier. *Invitation to fixed-parameter algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2006.

42 James B. Orlin. Contentment in graph theory: Covering graphs with cliques. *Indagationes Mathematicae (Proceedings)*, 80(5):406–424, 1977.

43 Hans-Peter Piepho. An algorithm for a letter-based representation of all-pairwise comparisons. *Journal of Computational and Graphical Statistics*, 13(2):456–466, 2004.

44 Subramanian Rajagopalan, Manish Vachharajani, and Sharad Malik. Handling irregular ILP within conventional VLIW schedulers using artificial resource constraints. In *CASES 2000*, pages 157–164, 2000. doi:10.1145/354880.354902.

45 Frank Plumpton Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society*, s2-30(1):264–286, 1930. doi:10.1112/plms/s2-30.1.264.

46 Herbert John Ryser. Intersection properties of finite sets. *Combinatorial Theory*, 14:79–92, 1973. doi:10.1016/0097-3165(73)90065-4.