Co-Degeneracy and Co-Treewidth: Using the Complement to Solve Dense Instances

Gabriel L. Duarte
Institute of Computing, Fluminense Federal University, Niterói, Brazil

Mateus de Oliveira Oliveira
Department of Informatics, University of Bergen, Norway

Uéverton S. Souza[1]
Institute of Computing, Fluminense Federal University,, Niterói, Brazil

Abstract

Clique-width and treewidth are two of the most important and useful graph parameters, and several problems can be solved efficiently when restricted to graphs of bounded clique-width or treewidth. Bounded treewidth implies bounded clique-width, but not vice versa. Problems like Longest Cycle, Longest Path, MaxCut, Edge Dominating Set, and Graph Coloring are fixed-parameter tractable when parameterized by the treewidth, but they cannot be solved in FPT time when parameterized by the clique-width unless FPT = W[1], as shown by Fomin, Golovach, Lokshtanov, and Saurabh [SIAM J. Comput. 2010, SIAM J. Comput. 2014]. For a given problem that is fixed-parameter tractable when parameterized by treewidth, but intractable when parameterized by clique-width, there may exist infinite families of instances of bounded clique-width and unbounded treewidth where the problem can be solved efficiently. In this work, we initiate a systematic study of the parameters co-treewidth (the treewidth of the complement of the input graph) and co-degeneracy (the degeneracy of the complement of the input graph). We show that Longest Cycle, Longest Path, and Edge Dominating Set are FPT when parameterized by co-degeneracy. On the other hand, Graph Coloring is para-NP-complete when parameterized by co-degeneracy but FPT when parameterized by the co-treewidth. Concerning MaxCut, we give an FPT algorithm parameterized by co-treewidth, while we leave open the complexity of the problem parameterized by co-degeneracy. Additionally, we show that Precoloring Extension is fixed-parameter tractable when parameterized by co-treewidth, while this problem is known to be W[1]-hard when parameterized by treewidth. These results give evidence that co-treewidth is a useful width parameter for handling dense instances of problems for which an FPT algorithm for clique-width is unlikely to exist. Finally, we develop an algorithmic framework for co-degeneracy based on the notion of Bondy-Chvátal closure.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Graph algorithms analysis; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases FPT, treewidth, degeneracy, complement graph, Bondy-Chvátal closure

Digital Object Identifier 10.4230/LIPIcs.MFCS.2021.42

Funding *Mateus de Oliveira Oliveira*: Bergen Research Foundation, and by the Research Council of Norway (Grant Number: 288761).
Uéverton S. Souza: Grant E-26/203.272/2017 - Carlos Chagas Filho Research Support Foundation (FAPERJ), and Grant 309832/2020-9 - National Council for Scientific and Technological Development (CNPq).

[1] Corresponding author

46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021).
Editors: Filippo Bonchi and Simon J. Puglisi; Article No. 42; pp. 42:1–42:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Treewidth and *clique-width* are two of the most important and useful graph parameters. Families of graphs of bounded treewidth include cactus graphs, outerplanar graphs, series-parallel graphs, Halin graphs, Apollonian networks [3], and graphs of bounded branch-width [45]. Graph classes with bounded clique-width include cographs [10], distance-hereditary graphs [35], and graphs of bounded treewidth [20]. Additionally, the clique-width of a graph is asymptotically equivalent to its rank-width [43].

An algorithmic meta-theorem due to Courcelle, Makowsky and Rotics [15] states that any problem expressible in the monadic second-order logic of graphs (MSO_1) can be solved in $f(cw) \cdot n$ time, i.e., it is fixed-parameter tractable when parameterized by the clique-width, cw, of the input graph. Originally this required a clique-width expression as part of the input. This restriction was removed when Oum and Seymour [43] gave an FPT algorithm, parameterized by the clique-width of the input graph, that finds a $2^{\mathcal{O}(cw)}$-approximation of an optimal clique-width expression. In addition, Courcelle [16] states that any problem expressible in the monadic second-order logic of graphs with edge set quantifications (MSO_2) can be solved in time $f(tw) \cdot n$, where tw is the treewidth of the input graph. Clearly, every MSO_1 property is also a MSO_2 property. However, there are MSO_2 properties like "G has an Hamiltonian cycle" that are not MSO_1 expressible [18]. These results have been extended a number of times [1, 9, 15, 37]. In particular, the MSO meta-theorems mentioned above were extended to LinEMSO by allowing the expressibility of optimization problems concerning maximum or minimum sets (LinEMSO properties are equivalent to MSO properties for optimization problems which can be expressed as searching for sets of vertices/edges that are optimal concerning some linear evaluation functions) [1, 17, 15]. From these meta-theorems, it follows that several problems can be efficiently solved when restricted to graphs of bounded clique-width or treewidth. Many optimization problems are LinEMSO_2-expressible, but Graph Coloring, the problem of determining the chromatic number of the input graph is not a LinEMSO problem [41]. However, on graphs G with bounded treewidth, the chromatic number of G is also bounded; therefore, in this specific case, the problem is Turing-reducible to a MSO_1 problem (k-Coloring for fixed k).

Bounded treewidth implies bounded clique-width [13] but the opposite implication is not valid, as in the case of complete graphs. On the other hand, LinEMSO_2 is more expressive than LinEMSO_1, and there exist LinEMSO_2-expressible problems like MaxCut, Longest Cycle, Longest Path and Edge Dominating Set that cannot be FPT when parameterized by clique-width [28, 29, 30, 31], unless FPT = W[1]. Additionally, Graph Coloring is also an FPT problem concerning treewidth parameterization that cannot be FPT when parameterized by clique-width, unless FPT = W[1], see [30].

For problems that are fixed-parameter tractable when parameterized by treewidth, but intractable when parameterized by clique-width, the identification of tractable classes of instances of bounded clique-width and unbounded treewidth becomes a fundamental quest. The goal of this work is to show that *co-treewidth*, that is to say, the treewidth of the complement of the input graph, is a parameter that fulfills this quest. More precisely, we will show that several natural problems that are unlikely to be in FPT when parameterized by clique-width can be solved in FPT time when parameterized by co-treewidth. Examples of such problems are Longest Path, Longest Cycle, MaxCut, Edge Dominating Set, and Graph Coloring. In addition, since bounded treewidth implies bounded degeneracy (the degeneracy of a graph is upper bounded by its treewidth), we also consider the degeneracy of the complement graph, called *co-degeneracy*, as a parameter.

Let us say that a parameter x is weaker than parameter y, and y stronger than x, if the set of graph classes where x is bounded is a subset of those where y is bounded. In 2016, Saether and Telle [46] considered a graph parameter called sm-width which is stronger than treewidth and weaker than clique-width. They showed that MaxCut, Graph Coloring, Hamiltonian Cycle and Edge Dominating Set are FPT when parameterized by sm-width. However, co-treewidth and sm-width are incomparable since trees have bounded sm-width but unbounded co-treewidth, and the complements of paths have bounded co-treewidth but unbounded sm-width. Also, note that neighborhood diversity [40], twin-cover [33], shrub-depth [34], and modular-width [32] are all weaker than clique-width, but none of them are stronger than co-treewidth. Gajarský, Lampis, and Ordyniak [32] showed that Graph Coloring and Hamiltonian Path are W[1]-hard parameterized by shrub-depth but FPT parameterized by modular-width (which is stronger than neighborhood diversity and twin-cover), they also leave as an open problem the complexity of MaxCut and Edge Dominating Set parameterized by modular-width. Besides, also in the context "between treewidth and clique-width", Eiben, Ganian, Hamm, and Kwon [26] develop hybrid parameters ($\mathcal{H}$-treewidth) combining advantages of treewidth and modulators, the aim of $\mathcal{H}$-treewidth is to capture the treewidth of a modulator to the class $\mathcal{H}$ (see [26]).

In 2016, Dvořák, Knop, and Masařík [25] showed that k-Path Cover is FPT when parameterized by the treewidth of the complement of the input graph (i.e., the co-treewidth of the input). This implies that Hamiltonian Path is FPT when parameterized by co-treewidth. In 2017, Knop, Koutecký, Masařík, and Toufar [38] asked about the complexity of deciding graph problems Π on the complement of G considering a parameter p of G (i.e., with respect to $p(G)$), especially for sparse graph parameters such as treewidth. In this paper, by showing that Longest Path, Longest Cycle, MaxCut, Edge Dominating Set, and Graph Coloring are all FPT when parameterized by co-treewidth, we exemplify that co-treewidth is a useful width parameter for dealing with problems for which an FPT algorithm for clique-width is unlikely. Besides, to the best of our knowledge, this is the first work dealing with co-degeneracy parameterization.

It is also natural to consider the clique-width of the complement graph as parameter, however, Courcelle and Olariu [20] proved that for every graph G we have $cw(\bar{G}) \leq 2 \cdot cw(G)$. Therefore, the notion of "co-clique-width" is redundant from the point of view of parameterized complexity. On the other hand, the notion of co-treewidth seems to be interesting given that bounded co-treewidth implies bounded clique-width; and treewidth and co-treewidth are incomparable parameters. Moreover, although co-degeneracy is incomparable with clique-width, it is stronger than co-treewidth and a useful parameter for handling some problems on dense instances. In this paper, we show that Longest Path, Longest Cycle, and Edge Dominating Set are FPT when parameterized by co-degeneracy, while Graph Coloring is para-NP-hard. The complexity of MaxCut parameterized by co-degeneracy is left open.

Finally, we also remark that for some graph problems, co-treewidth can be a parameter more useful than treewidth. For instance, Equitable Coloring and Precoloring Extension are well-known W[1]-hard problems concerning treewidth; however, we remark that both problems are fixed-parameter tractable when parameterized by co-treewidth.

1.1 Preliminaries

We use standard graph-theoretic and parameterized complexity notation, and we refer the reader to [21, 24] for any undefined notation.

The degeneracy of a graph G is the least k such that every induced subgraph of G contains a vertex with degree at most k. Equivalently, the degeneracy of G is the least k such that its vertices can be arranged into a sequence so that each vertex is adjacent to most

k vertices preceding it in the sequence. We denote by co-deg(G) the co-degeneracy of G, i.e., the degeneracy of $\overline{G}$. Also, for a graph G, we denote by co-tw(G) the co-treewidth of G, i.e., the treewidth of $\overline{G}$. For short, we use *co-deg* and *co-tw* whenever the graph G is implicit.

In general, for a tree decomposition $(T, \{X_t\}_{t \in V(T)})$ it is common to distinguish one vertex r of T which will be the root of T. This introduces natural parent-child and ancestor-descendant relations in the tree T [21]. It is useful to design dynamic programming algorithms based on tree decompositions to obtain rooted tree decompositions that satisfy some auxiliary conditions. Such decompositions are so-called *nice tree decompositions*.

▶ **Definition 1.** *A tree decomposition $(T, \{X_t\}_{t \in V(T)})$ with* root *node X_r is nice, if the following conditions are satisfied:*

- *$X_r = \emptyset$; (the root is an empty bag of T)*
- *If X_ℓ is a* leaf *node of T, then $X_\ell = \emptyset$; (each leaf X_ℓ is an empty bag of T)*
- *Every non-leaf node of T is of one of the following three types:*
  1. Introduce *node: a node t with exactly one child t' such that $X_t = X_{t'} \cup \{v\}$ for some vertex $v \notin X_{t'}$; we say that v is* introduced *at t.*
  2. Forget *node: a node t with exactly one child t' such that $X_t = X_{t'} \setminus \{w\}$ for some vertex $w \in X_{t'}$; we say that w is* forgotten *at t.*
  3. Join *node: a node t with two children t_1, t_2 such that $X_t = X_{t_1} = X_{t_2}$.*

Let G be a graph and let $(T, \{X_t\}_{t \in V(T)})$ be a nice tree-decomposition of the graph G. For each node t of T, we denote by T_t the subtree of T rooted by t. Also, we denote by G_t the subgraph of G induced by the set of vertices contained in some bag of T_t.

Based on the following results, we can assume that we are given a nice tree decomposition of G without loss of generality.

▶ **Theorem 2** ([5]). *There exists an algorithm that, given an n-vertex graph G and an integer k, runs in time $2^{O(k)} \cdot n$ and either outputs that the treewidth of G is larger than k, or constructs a tree decomposition of G of width at most $5k + 4$.*

▶ **Lemma 3** ([21]). *Given a tree decomposition $(T, \{X_t\}_{t \in V(T)})$ of G of width at most k, one can in time $O(k^2 \cdot \max(|V(T)|, |V(G)|))$ compute a nice tree decomposition of G of width at most k that has at most $O(k \cdot |V(G)|)$ nodes.*

The *clique-width* of a graph is another parameter that we will mention very often, and therefore, we briefly define this parameter for completeness. Given a graph G, the clique-width of G, denoted by $cw(G)$, is defined as the minimum number of labels needed to construct G, using the following four operations: create a single vertex v with an integer label ℓ (denoted by $\ell(v)$); take the disjoint union (i.e., co-join) of two graphs (denoted by $\oplus$); join by an (arc) edge every vertex labeled i to every vertex labeled j for $i \neq j$ (denoted by $\eta(i, j)$); relabel all vertices with label i by label j (denoted by $\rho(i, j)$). An algebraic term representing such a construction of G and using at most k labels is a *k-expression* of G. The clique-width of G is the minimum k for which G has a k-expression.

Given a graph G and a vertex $v \in V(G)$, $N(v)$ denotes the (open) neighborhood of v, $N[v]$ denotes the closed neighborhood of v ($N[v] = N(v) \cup \{v\}$), and $d(v)$ denotes the degree of v ($d(v) = |N(v)|$).

We say that two vertices v, w of G have the same type if $N(v) \setminus \{w\} = N(w) \setminus \{v\}$.

A graph G has neighborhood diversity at most k, if there exists a partition of $V(G)$ into at most k sets, such that all the vertices in each set have the same type. We denote by $nd(G)$ (or just nd when the graph G is implicit) the least k such that G has neighborhood diversity at most k.

A set $S \subseteq V(G)$ is a *vertex cover* of a graph G if for each edge of G at least one of its endpoints is in S. The *vertex cover number* of G, denoted by $vc(G)$, is the least k such that G has a vertex cover of size k. It is well known that if a graph G has vertex cover at most k, then $nd(G) \leq 2^k + k$ (c.f. [40]).

A *path cover* $\mathcal{P}$ of a graph G is a set of vertex-disjoint paths of G such that each vertex in $V(G)$ belongs to a path in $\mathcal{P}$.

A graph G is *Hamiltonian* if there is a cycle which includes every vertex of G (such a cycle is called a *Hamiltonian cycle*). A graph G is said k-Hamiltonian if the deletion of at most k vertices from G results in a Hamiltonian graph.

Finally, we denote by n the number of vertices of the graph under consideration.

2 Monadic second-order logic for graphs with bounded co-treewidth

Dvořák, Knop, and Masařík [25] asked whether it is possible to extend the meta-theorem for MSO_2 for the complementary setting – i.e. to allow quantification over sets of non-edges. As shown by Courcelle, Makowsky and Rotics [15] (assuming P $\neq$ NP on unary languages), it is not possible to allow quantification over sets of edges as well as quantification over sets of non-edges, under the requirement that for the target parameter the complete graphs should have bounded width. However, as observed by Knop, Koutecký, Masařík, and Toufar [38], the result that k-Path Cover[2] is FPT when parameterized by co-treewidth suggests that at least sometimes some extension of MSO theorem can be useful to decide properties of the complement graph. Next, by way of illustration, we show that this is precisely the case of Balanced co-Biclique, the problem of determining the maximum integer k for which the input graph G has a pair of cliques K_1, K_2 such that $|K_1| = |K_2| = k$ and there is no edge from K_1 to K_2. Such a pair of cliques is the complement of a balanced complete bipartite graph (balanced biclique).

Let $\text{LinEMSO}_{\bar{2}}$ be the extension of LinEMSO_1, where quantification is allowed over sets of non-edges, but quantification over sets of edges is not allowed. It is easy to see that $\text{LinEMSO}_{\bar{2}}$-expressible problems are FPT concerning co-treewidth since LinEMSO_2-expressible problems are FPT concerning treewidth. Note that expressing a property in $\text{LinEMSO}_{\bar{2}}$ is equivalent to expressing the complementary property in LinEMSO_2.

▶ **Lemma 4.** *Finding the maximum balanced biclique of a graph is LinEMSO_2-expressible.*

Proof. Given a graph G and set of vertices S_1, it is easy to express in MSO_1 the existence of a disjoint set S_2 such that S_1 and S_2 form an induced complete bipartite subgraph: S_1 and S_2 must be independent sets, and G must contain all possible edges from S_1 to S_2. Although comparing the cardinality of sets is not allowed in LinEMSO_2, in this particular case, it would still be possible by verifying the existence of a perfect matching in the subgraph induced by $S_1 \cup S_2$. Thus, the problem of finding the largest S_1 meeting these conditions is LinEMSO_2-expressible. ◀

From Lemma 4 and the LinEMSO_2 meta-theorem for optimization problems parameterized by treewidth [19], it follows that Corollary 5 holds.

▶ **Corollary 5.** Balanced co-Biclique *is fixed-parameter tractable when parameterized by the co-treewidth of G.*

[2] k-Path Cover is the problem of finding a path cover of size k, where k is fixed.

3 Bondy-Chvátal closure, stability and co-degeneracy

Let G be a graph with n vertices and let u and v be distinct nonadjacent vertices of G such that $d(u) + d(v) \geq n$. Ore's theorem states that G is hamiltonian if and only if $G + uv$ is hamiltonian. In 1976, Bondy and Chvátal [8] generalized Ore's theorem and defined a helpful tool: the *closure* of a graph.

Let ℓ be an integer. The $(n + \ell)$-closure $cl_{n+\ell}(G)$ of a graph G is obtained from G by recursively adding an edge between pairs of nonadjacent vertices whose degree sum is at least $n + \ell$ until no such pair remains. Bondy and Chvátal showed that $cl_{n+\ell}(G)$ is uniquely determined by G and that G is hamiltonian if and only if $cl_n(G)$ is hamiltonian.

First, observe that the classes of graphs with bounded co-degeneracy and bounded co-treewidth are both closed under completion (edge addition), just as bounded degeneracy and treewidth are closed under edge removals. Therefore, regarding Hamiltonian Cycle on graphs with co-degeneracy or co-treewidth k, without loss of generality, we can assume that $G = cl_n(G)$.

Dvořák, Knop and Masařík [25] showed that if a graph G has co-treewidth k and $G = cl_n(G)$ (that is, closed under Bondy-Chvátal closure) then G has neighborhood diversity bounded by $2^{2(k^2+k)} + 2(k^2 + k)$. Below we present some stronger results.

We call by *co-vertex cover* any set of vertices whose removal makes the resulting graph complete, i.e., a vertex cover in the complement. The *co-vertex cover number*, *co-vc*(G), of a graph G is the minimum cardinality of a co-vertex cover in G. Recall that *co-vc*(G) is also called *distance to clique*, and a co-vertex cover set is also called a *clique modulator*.

▶ **Theorem 6.** *Let $\ell \geq 0$ be an integer. If a graph G has co-degeneracy k and $G = cl_{n+\ell}(G)$ then G has co-vertex cover number bounded by $2k + \ell + 1$. In addition, a co-vertex cover of G with size at most $2k + \ell + 1$ can be found in polynomial time.*

Proof. Let G be a graph such that $G = cl_{n+\ell}(G)$ and co-deg(G)=k.

We may assume that G has at least $2k + \ell + 2$ vertices.

Let $v_1, v_2, v_3, \ldots, v_n$ be an ordering of the vertices of $\overline{G}$ obtained by repeatedly removing the minimum-degree vertex of $\overline{G}$. For each $t \in \{1, 2, \ldots, n\}$ we denote by G^t the subgraph of G induced by $\{v_i : 1 \leq i \leq t\}$.

Note that G^1 is a complete graph (i.e., a K_1). Therefore, $t = 1$ is the base case.

Now, let t be an integer such that $2 \leq t \leq n$ and $|V(G) \setminus V(G^t)| \geq 2k + \ell + 1$.

Suppose by hypothesis that G^{t-1} is a complete graph. At this point, it remains to prove that G^t is also a complete graph.

Since $V(G^{t-1}) = \{v_1, \ldots, v_{t-1}\}$ is a clique of G and co-deg(G)=k, it holds that each vertex of $V(G^{t-1})$ has degree at least $n - k - 1$ in G. Also, the vertex v_t has at least $k + \ell + 1$ neighbors in $V(G) \setminus V(G^t)$, because $|V(G) \setminus V(G^t)| \geq 2k + \ell + 1$, and by the co-degeneracy the vertex v_t has at most k non-neighbors in $V(G) \setminus V(G^t)$. Finally, as $G = cl_{n+\ell}(G)$ it holds that v_t is adjacent to all vertices of G^{t-1} in the graph G, which implies that $V(G^t)$ is also a clique of G.

Thus, in a left-right manner according the ordering, $v_1, v_2, v_3, \ldots, v_n$, we can observe that each $V(G^t)$ induces a clique until meeting the first vertex v_j such that $|V(G) \setminus V(G^j)| < 2k + \ell + 1$. This implies that $\{v_j, v_{j+1}, \ldots, v_n\}$ is a co-vertex cover of G with size at most $2k + \ell + 1$, and it can be computed in polynomial time. ◀

As a corollary we improve the Dvořák-Knop-Masařík bound with respect to neighborhood diversity for $\ell = 0$.

▶ **Corollary 7.** *Let* $\ell \geq 0$ *be an integer. If a graph* G *has co-degeneracy* k *and* $G = cl_{n+\ell}(G)$ *then* G *has neighborhood diversity bounded by* $2^{2k+\ell+1} + 2k + \ell + 1$*.*

In [25], from the fact that if $G = cl_n(G)$ then the neighborhood diversity is bounded by a function of the co-treewidth, it is claimed that Hamiltonian Path is in FPT concerning co-treewidth. For an FPT algorithm for Hamiltonian Path parameterized by the neighborhood diversity (nd), they point to the seminal paper of Lampis [40], which uses an algorithm of Cosmadakis and Papadimitriou [14] resulting in solving the problem in time $O^*(2^{nd \cdot \log nd})$. Recall that this strategy implies a solution for Hamiltonian Path in double exponential time with respect to the co-treewidth. Below we present a much more efficient algorithm. First, we consider co-vertex cover number parameterization.

▶ **Lemma 8.** Longest Path *and* Longest Cycle *can be solved in time* $2^{O(k \cdot \log k)} \cdot n^{O(1)}$ *where* k *is the co-vertex cover number of the input graph.*

Proof. Let S, K be a partition of the vertices of a graph G into a co-vertex cover S and a clique K, where $|S| = k$. We assume that $|K| > 2|S|$; otherwise we can "guess", in single-exponential time, the vertices in the longest path/cycle, so one can solve both problems using single-exponential exact algorithms for Hamiltonian Cycle (or TSP), such as the Bellman–Held-Karp algorithm [2, 36].

Since $|K| > 2|S|$, there is a longest cycle and a longest path containing all vertices of K, otherwise any longest cycle/path either has no edge between two vertices of K (therefore, it has size at most $2|S|$), or it has an edge uv where $u, v \in K$, implying that there is a larger cycle/path obtained by replacing uv by a uv-path containing as internal vertices the vertices of K that were not in the cycle/path, both cases contradict the fact that the cycle/path is the longest. Also, note that Longest Path can be reduced to Longest Cycle by adding one universal vertex. Therefore, we focus on Longest Cycle.

Now, in time $2^{O(|S|)}$ one can branch by guessing the set S_x of vertices of S that are not in the longest cycle, and then removing S_x. After that, we may assume that we are dealing with an instance G' of the Hamiltonian Cycle problem, where $V' = V(G) \setminus S_x$, $G' = G[V']$, $K' = K$, and $S' = S \setminus S_x$ is a co-vertex cover of G'.

Let k' be the cardinality of S'. Recall that $k' \leq k$. Now, we branch by guessing a permutation $s_1, s_2, \ldots, s_{k'}$ of the vertices of S' representing a circular order of visits of the vertices of S' in the Hamiltonian cycle C (if any). Given such a permutation $s_1, s_2, \ldots, s_{k'}$, we guess the edges s_i, s_{i+1} of G' that are in C. Note that these branching steps take $O(k! \cdot 2^k)$ time. Recall that $O(k! \cdot 2^k) = 2^{O(k \log k)}$. At this point, we have guessed the set of subpaths of C induced by S'. Let $P_1, P_2, \ldots, P_r$ be the circular order of visits of such paths according to the guessed permutation.

For each pair of consecutive paths P_i, P_{i+1} either their corresponding endpoints are connected by a common neighbor in C or there is a path of vertices of K between them in such a cycle. Again, we branch by guessing in time $2^{O(k)}$ the pairs connected by a common neighbor. After that, we can construct a bipartite graph B with bipartiton $V(B) = (B_1, B_2)$ where: each vertex of B_1 represents either a pair of endpoints of the paths that must have a common neighbor in C, or an endpoint that has a distinct neighbor inside K along C; B_2 is the set of vertices of the clique K, and $E(B)$ is defined according to the edges from S' to K (for vertices representing pairs of endpoints, the neighborhood is the vertices of K that are neighbors of both endpoints). Clearly, if the guessed structures are feasible for obtaining a Hamiltonian cycle C then B has a matching of size $|B_1|$. Let M be such a matching, if any.

Since $|K| > 2|S|$ we assume that for at least one pair P_i, P_{i+1} its corresponding endpoints do not have a common neighbor in C. Hence, having the matching M and such a path cover $P_1, \ldots, P_r$ of $G'[S']$, a Hamiltonian cycle can be easily obtained for G'. Therefore, Longest Cycle and Longest Path can be solved in time $2^{O(k \log k)} \cdot n^{O(1)}$, where $k = co\text{-}deg(G)$. ◀

Since Hamiltonian Path can be reduced to Hamiltonian Cycle by adding a universal vertex, from Bondy-Chvátal Theorem, Theorem 6 and Lemma 8, the following holds.

▶ **Corollary 9.** Hamiltonian Cycle *and* Hamiltonian Path *can be solved in time* $2^{O(k \log k)} \cdot n^{O(1)}$*, where* k *is the co-degeneracy of the input graph.*

In order to extend the previous result to Path Cover, Longest Cycle, and Longest Path as well as considering the same strategy for other properties, it becomes necessary to introduce the notion of stability.

3.1 The stability of a property

A property P defined on all graphs of order n is said to be $(n+\ell)$-stable if for any graph G of order n that does not satisfy P, the fact that uv is not an edge of G and that $G+uv$ satisfies P implies $d(u)+d(v) < n+\ell$. In other words, if $uv \notin E(G)$, $d(u)+d(v) \geq n+\ell$ and $G+uv$ has property P, then G itself has property P (c.f. [12]). We denote by $s(P)$ the smallest integer $n+\ell$ such that P is $(n+\ell)$-stable, and call it the *stability* of P.

Note that if a graph property P is $(n+\ell)$-stable then edges between pair of vertices u, v such that $d(u)+d(v) \geq n+\ell$ can be added without destroying such a property P.

Our co-degeneracy+closure+co-vertex cover framework is based on the following facts:

1. If a property P is $(n+\ell)$-stable and $cl_{n+\ell}(G)$ satisfies P, then G itself satisfies P.
2. If a property P is $(n+\ell)$-stable, regarding the problem of recognizing property P parameterized by co-degeneracy we can assume without loss of generality that $G = cl_{n+\ell}(G)$.
3. If $G = cl_{n+\ell}(G)$ and G has co-degeneracy k then G has a co-vertex cover of size at most $2k+\ell+1$. In particular, we are interested in cases where ℓ is bounded by a function of k.
4. Many problems are easily solved in FPT-time concerning co-vertex cover parameterization.

Next, we list the stability of some graph properties P (see [8, 11, 12]):

Longest Cycle: "G has circumference k" satisfies $s(P) = n$. (Thm. 4 in [11])
Longest Path: "G contains a P_k" satisfies $s(P) = n-1$ for $4 \leq k \leq n$. (Thm. 2.40 in [12])
Path Cover: "G has a path cover of size at most k" satisfies $s(P) = n-k$. (Thm. 9.13 in [8])
k-Hamiltonian: "G is k-Hamiltonian" satisfies $s(P) = n+k$. (Thm. 2.25 in [12])

At this point, it is easy to see that Corollary 10 holds.

▶ **Corollary 10.** Longest Cycle/Path *can be solved in time* $2^{O(co\text{-}deg \log co\text{-}deg)} \cdot n^{O(1)}$*, and* k*-Hamiltonian graphs can be recognized in time* $2^{O((co\text{-}deg+k)\log(co\text{-}deg+k))} \cdot n^{O(1)}$*.*

Proof. By the stability of the properties regarding the computation of the longest cycle, longest path, and recognition of k-Hamiltonian graphs, it holds that one can assume that G is closed under Bondy-Chvátal closure for an appropriated integer ℓ ($G = cl_{n+\ell}(G)$). By Theorem 6, it holds that G has a co-vertex cover S of size $O(\text{co-deg})$, or $O(\text{co-deg}+k)$ in the k-Hamiltonian case, and such a co-vertex cover can be obtained in polynomial time.

Given a co-vertex cover S of G, in time $2^{O(|S|)}$ one can "guess" the vertices of S that must be removed or are not in the longest cycle/path. After that, the proof follows as in Lemma 8. ◀

Next, we deal with the Path Cover problem.

▶ **Corollary 11.** Path Cover *can be solved in time* $2^{O(co\text{-}deg \log co\text{-}deg)} \cdot n^{O(1)}$*.*

Proof. By the stability of the property P of having a path cover of size at most k (see [8]), without loss of generality, we can assume $G = cl_{n-k}(G)$. Note that $E(G) \subseteq E(cl_n(G)) \subseteq E(cl_{n-k}(G))$. Thus, from Theorem 6 it holds that G has co-vertex cover number at most $2k+1$, where $k = \textit{co-deg}(G)$. In addition, such a co-vertex cover S can be found in polynomial time. Thus, G has a path cover of size at most $2k+2$, because one can use a trivial path for each vertex of the co-vertex cover, and a single path for the remaining vertices (they induce a clique). So, the path cover number of a graph G is bounded by $2k+2$. At this point, it is enough to determine the least $r \in [1, 2k+2]$ for which G has an r-path cover (path cover of size r). Since one can reduce the problem of finding an r-path cover to Hamiltonian Path by adding $r-1$ universal vertices, and the addition of universal vertices preserve the co-vertex cover number of the input graph, by Corollary 9 it holds that Path Cover can also be solved in time $2^{O(k \log k)} \cdot n^{O(1)}$, where $k = \textit{co-deg}(G)$. ◀

4 Edge Dominating Set parameterized by co-degeneracy

An edge dominating set of a graph G is a set $Q \subseteq E(G)$ such that every edge of G is either in Q or incident to at least one edge of Q. The Edge Dominating Set problem consists of determining the size of a minimum edge dominating set.

In [30], Fomin et al. showed that Edge Dominating Set parameterized by clique-width is W[1]-hard. In [29, 31], they showed that Edge Dominating Set cannot be solved in time $f(cw) \cdot n^{o(cw)}$, unless ETH fails.

In this section, we present a single-exponential FPT algorithm for Edge Dominating Set parameterized by the co-degeneracy.

▶ **Theorem 12.** Edge Dominating Set *can be solved in time* $2^{O(\textit{co-deg})} \cdot n^{O(1)}$.

Proof. First, we observe the following key property.

▷ Claim 13. The problem of finding a minimum edge dominating set is equivalent to finding the smallest vertex cover S such that $G[S]$ contains a perfect matching.

Proof. Yannakakis and Gavril [49] showed that given a minimum edge dominating set Q of G, one could find a minimum maximal matching with $|Q|$ edges. Since every maximal matching is an edge dominating set, the size of a minimum edge dominating set equals the size of a minimum maximal matching.

Now, let Q be a minimum edge dominating set that is also a minimum maximal matching. Since Q is a maximal matching then $V(Q)$ is a vertex cover inducing a graph having perfect matching. In addition, there is no vertex cover smaller than $V(Q)$ that also induces a graph having perfect matching; otherwise, Q is not a minimum maximal matching.

Conversely, let S be the smallest vertex cover S of G such that $G[S]$ contains a perfect matching. Let Q be such a perfect matching of $G[S]$. Since S is a vertex cover of G then $V \setminus S$ is an independent set, which implies that Q is a maximal matching of G. In addition, Q must be a minimum maximal matching, otherwise using the previous argument we obtain a vertex cover smaller than S also having a perfect matching, a contradiction. ◁

Now, recall that enumerating vertex covers in G is the same as enumerating independent sets in the same graph, which is equivalent to enumerating the cliques of $\overline{G}$.

Since $\overline{G}$ has degeneracy k, we can enumerate all cliques containing some vertex of degree at most k (such a vertex must exist and there are at most 2^k cliques containing it); by deleting this vertex and continuing the enumeration in the remaining graph, we can enumerate every clique of $\overline{G}$.

Therefore, if a graph G has co-degeneracy k, then G has at most $2^k \cdot n$ distinct vertex covers which we can enumerate in time $O(2^k \cdot n^{O(1)})$. Thus, by checking the existence of perfect matchings, one can find the minimum edge dominating set in time $2^k \cdot n^{O(1)}$. ◀

5 MaxCut parameterized by co-treewidth

A cut $[S, V \setminus S]$ of a graph $G = (V, E)$ is a partition of V into two subsets S and $V \setminus S$. The *size* of the cut $[S, V \setminus S]$ is the number of edges crossing it, i.e., the cardinality of the cut-set $\{uv \in E \mid u \in S, v \in V \setminus S\}$. In the MaxCut problem, we are given an unweighted undirected graph $G = (V, E)$, and our goal is to find a cut of maximum size.

From the parameterized complexity point of view, to determine if G has a cut of size at least k is fixed-parameter tractable when parameterized by either k or $k - |\frac{|E|}{2}|$ (c.f. [42, 44]). In addition, in 2000, Bodlaender and Jansen [6] showed that MaxCut can be solved in FPT time when parameterized by the treewidth, and, in 2013, Bodlaender, Bonsma, and Lokshtanov presented an $O(2^{tw} \cdot n)$ time algorithm for the problem, where tw is the treewidth of the input graph. On the other hand, Fomin, Golovach, Lokshtanov and Saurabh [29, 31] showed that MaxCut cannot be solved in time $f(cw) \cdot n^{o(cw)}$, unless ETH fails, where cw is the clique-width of the input graph G.

Regarding co-degeneracy parameterization we left the complexity of MaxCut open. To the best of our knowledge, the complexity of MaxCut is unknown even for co-planar graphs (the class of planar graphs is a subclass of the 5-degenerate graphs).

Concerning co-treewidth, it is not clear whether MaxCut can be expressed in $\text{LinEMSO}_{\overline{2}}$. Given a cut $[S, V \setminus S]$ of a graph G, the complement of the cut-set of $[S, V \setminus S]$ is the set of non-edges that have one endpoint in each subset of the partition. The main challenge to express MaxCut using just quantification over sets of vertices and sets of non-edges is that the size of $[S, V \setminus S]$ is given by $(|S| \cdot |V \setminus S|) - |\{uv \notin E \mid u \in S, v \in V \setminus S\}|$, thus, the natural objective function is not linear, and it is not appropriate to express the problem in LinEMSO. However, using a nice tree decomposition of the complement graph, we can find the maximum cut of the input graph in single-exponential time concerning the co-treewidth.

Given a graph G and a nice tree-decomposition $(T, \{X_t\}_{t \in V(T)})$ of G, our goal is to use $(T, \{X_t\}_{t \in V(T)})$ in order to find a maximum cut of $\overline{G}$. Recall that for each node t of T, we denote by G_t the subgraph of G induced by the set of vertices contained in some bag of T_t. Also, for each X_t the set of forgotten vertices in G_t is denoted by F_t ($F_t = V(G_t) \setminus X_t$).

Given a cut $[S, V \setminus S]$, we say that S is left part of the partition defined by the cut, and $V \setminus S$ is the right part.

Let $C[t, S, \ell]$ be the size of a maximum cut of the subgraph $\overline{G_t}$, where S are the vertices of X_t to the left part of the partition defined by the cut, and $X_t \setminus S$ are the vertices of X_t on the right part. Also, ℓ represents how many forgotten vertices are in the left part of this cut.

At this point, it is sufficient to show how to compute in a bottom-up manner the entries of the matrix (which is regarding $\overline{G}$ (not G)) according to each type of node of the nice tree decomposition of G. Note that the maximum size of a cut in $\overline{G}$ equals $\max_{0 \le \ell \le n}\{C[r, \emptyset, \ell]\}$, where r is the root of the tree decomposition.

For each node t of T we denote by t' and t'' the children of t (if any), and for each bag X_t we denote by $E(X_t)$ the set of edges with both endpoints in X_t. Thus, we proceed as follows:

Leaf: Since $X_t = \emptyset$, it holds that $S = \emptyset$, $\ell = 0$, and $C[t, S, \ell] = 0$.

Introduce vertex: $X_t = X_{t'} \cup \{v\}$. Recall that we are working with the tree decomposition of G, so the size of cuts of $\overline{G}$ is given by non-edges of G.

$$C[t,S,\ell] = \begin{cases} C[t', S\setminus\{v\}, \ell] + |X_t \setminus S| - |\{uv \in E(X_t) : u \notin S\}| + (|F_{t'}| - \ell) & \text{if } v \in S \\ C[t', S, \ell] + |S| - |\{uv \in E(X_t), u \in S\}| + \ell & \text{if } v \notin S \end{cases} \quad (1)$$

Note that $|F_{t'}| - \ell$ is the number of vertices forgotten to the right part of the cut.

Forget vertex: $X_t = X_{t'} \setminus \{v\}$. In this case, we take the best of two possibilities: v in the left or right part of the cut.

$$C[t, S, \ell] = \max\{C[t', S \cup \{v\}, \ell - 1], C[t', S, \ell]\} \quad (2)$$

Join: $X_t = X_{t'} = X_{t''}$. In this case we have to do the union of two partial solutions. Since non-edges of $G[X_t]$ are non-edges of both $G_{t'}$ and $G_{t''}$, they must not be counted twice. In addition, there are non-edges between forgotten vertices of $G_{t'}$ and $G_{t''}$ that must be counted, so:

$$C[t,S,\ell] = \max_{0\leq i\leq \ell}\{C[t',S,i] + C[t'',S,\ell-i] - (|S|\cdot|X_t\setminus S| - |\{uv \in E(X_t) : u \in S, v \notin S\}|) + i\cdot(|F_{t''}| - (\ell - i)) + (|F_{t'}| - i)\cdot(\ell - i)\} \quad (3)$$

Since the correctness of the recurrences is straightforward, the matrix has size $2^{\mathcal{O}(\text{co-tw})} \cdot n^2$, and each entry can be computed in time $\mathcal{O}(n)$, the following theorem holds.

▶ **Theorem 14.** MaxCut *can be solved in* $2^{\mathcal{O}(\textit{co-tw})} \cdot n^3$.

6 Treewidth vs. co-treewidth

In the previous sections we showed that Path Cover, Longest Cycle, Longest Path, MaxCut, and Edge Dominating Set are all FPT concerning co-treewidth parameterization. These results contrast with the intractability of such problems regarding clique-width parameterization. Since all of these problems are also fixed-parameter tractable when parameterized by treewidth, it becomes interesting to identify problems that are tractable for co-treewidth but intractable concerning treewidth, as well as the opposite.

The TSP problem is NP-hard on complete graphs (co-treewidth equal to zero) but fixed-parameter tractable when parameterized by treewidth [4]. On the other hand, Fellows et al. [27] showed that Precoloring Extension and Equitable Coloring are W[1]-hard when parameterized by treewidth; next, we contrast these results by remarking that both problems are fixed-parameter tractable using co-treewidth as the parameter.

6.1 Coloring and covering problems

Each color class of a proper coloring of a graph G is an independent set, i.e., each color class is a clique in the complement. So, to solve Graph Coloring working with the complement graph, we must solve Clique Cover in $\overline{G}$. Therefore, Graph coloring parameterized by co-degeneracy/co-treewidth is equivalent to Clique Cover parameterized by degeneracy/treewidth. It is known that Graph coloring and Clique Cover are NP-hard on planar graphs [39, 22]. Thus, they are para-NP-hard with respect to co-degeneracy.

Regarding co-treewidth, it is not clear if Clique Cover can be solved using the $MSOL_2$ framework because in contrast with Graph Coloring, the size of the solution is unbounded on bounded treewidth graphs. However, in [47], van Rooij, Bodlaender, van Leeuwen, Rossmanith, and Vatshelle present an FPT algorithm for γ-Clique Cover parameterized by treewidth. The γ-Clique Cover problem is a generalization of Clique Cover where the goal is to find a minimum collection C of disjoint cliques covering $V(G)$ such that the size of every clique in C is contained in γ (a set of integers). They showed a FPT algorithm that computes the size and number of minimum γ-clique covers of G. Preliminary parts of [47] have appeared in [7, 48]. Thus, Graph Coloring is FPT concerning co-treewidth.

The Equitable Coloring problem is a variation of Graph Coloring where we are asked to find the minimum integer k for which the input graph G admits a proper k-coloring such that the sizes of any two color classes differ by at most one. Again, to solve Equitable Coloring one can consider the complementary problem, i.e., Equitable Clique Cover parameterized by treewidth. In the Equitable Clique Cover problem, we are asked to find the minimum integer k such that the input graph G admits a clique cover of size k such that the sizes of any two cliques of the cover differ by at most one. Thus, one can use the algorithm for γ-Clique Cover to solve Equitable Clique Cover, considering that all the cliques must have size either ℓ or $\ell - 1$ ($\gamma = \{\ell, \ell - 1\}$). Using the folklore fact that for a graph G, every clique of G is contained in some bag of a tree decomposition of G, it follows that we only need to consider ℓ in $[2, tw(G) + 1]$. Therefore, the running time of the algorithm increases by a factor of at most $tw(G)$, and Equitable Coloring can also be solved in FPT time concerning co-treewidth. Besides, Gomes, Lima and dos Santos [23], using fast subset convolution as in [47], also showed an FPT-algorithm concerning treewidth for counting clique covers of G having only cliques of size ℓ and $\ell - 1$.

6.1.1 Precoloring Extension parameterized by co-treewidth

Precoloring Extension is a generalization of Graph Coloring, where we are given a graph $G = (V, E)$ with a subset $P \subseteq V$ of precolored vertices, a precoloring c_P of the vertices of P, and asked to determine the minimum integer k for which G admits a proper k-coloring c which extends c_P (that is, $c(v) = c_P(v)$ for all $v \in P$).

Again, we work with the complementary problem, which we propose to call Clique Cover Extension. In such a problem the input is the same as in Precoloring Extension, and the goal is to determine the minimum size of a clique cover for which vertices with the same color are in the same clique, and no clique has a pair of precolored vertices v, u such that $c_P(v) \neq c_P(u)$.

Next, we present a standard dynamic programming based on nice tree decompositions to solve Clique Cover Extension parameterized by treewidth. The proposed algorithm has a single-exponential dependency on the treewidth and preserves linearity with respect to n.

First, we assume that each color class induces a clique; otherwise, there is no solution. Thus, in the forget node t of a precolored vertex v it holds that all vertices precolored with color $c_P(v)$ belong to the graph G_t.

Let $C[t, S]$ be the minimum number of cliques needed to cover the vertices of $V(G_t) \setminus S$ in G_t (the subgraph rooted by the node t) according to the constraints of Clique Cover Extension, where S is a subset of X_t. Since each clique is contained in some bag, we assume that each clique is formed when its last vertex is introduced. Therefore, the matrix is filled in a bottom-up manner as follows.

Leaf: $X_t = \emptyset$, thus $S = \emptyset$ and $C[t, S] = 0$.

Introduce vertex: We are introducing the vertex v and all its edges. Thus, we have two possibilities: $v \in S$ or $v \notin S$. In the second case, v forms a new clique which either has size one or is formed by v together with some non-covered neighbors in $G_{t'}$.

$$C[t, S] = \begin{cases} C[t', S \setminus \{v\}], & \text{if } v \in S \\ \min_{\forall W}\{C[t', S \cup W]\} + 1, & \text{if } v \notin S \end{cases} \tag{4}$$

where the minimum is taken over all possible $W \subseteq N(v) \cap (X_t \setminus S)$ such that $G[W]$ is a clique (including the empty set) and:

- $W \cup \{v\}$ contains no pair of precolored vertices u, w such that $c_P(u) \neq c_P(w)$;
- if $W \cup \{v\}$ contains a precolored vertex then it contains all vertices precolored with the same color.

Forget vertex: In this node the vertex v is forgotten. Since this vertex must be covered with a clique of $G_{t'}$, we have the following

$$C[t, S] = C[t', S]$$

Join: In this case, we are joining solutions of the graphs rooted by nodes $X_{t'}$ and $X_{t''}$. Every vertex of $V(G_t) \setminus S$ should be covered by a clique in either $G_{t'}$ or $G_{t''}$. To avoid counting twice some cliques of $G[X_t]$, it is sufficient to note that if a vertex of X_t is covered in $G_{t'}$, then we can assume that it is not covered in $G_{t''}$, and vice versa. This implies that for each pair of solutions to be analyzed (one from each child), the cliques of $G[X_t]$ are considered in at most one of them.

$$C[t, S] = \min_{\forall S', S''}\{C[t', S'] + C[t'', S'']\}$$

where the minimum is taken over all possible S', S'' such that $S = S' \cap S''$, $S' \cup S'' = X_t$.

Since the matrix has size $\mathcal{O}(2^{tw(G)} \cdot tw(G) \cdot n)$, and each entry can be computed in $\mathcal{O}(2^{tw(G)})$, the following holds.

▶ **Theorem 15.** CLIQUE COVER EXTENSION *can be solved in* $2^{\mathcal{O}(tw(G))} \cdot n$.

▶ **Corollary 16.** PRECOLORING EXTENSION *can be solved in* $2^{\mathcal{O}(co\text{-}tw(G))} \cdot n$.

7 Concluding Remarks

LONGEST CYCLE, LONGEST PATH, PATH COVER, MAXCUT, EDGE DOMINATING SET and GRAPH COLORING are all fixed-parameter tractable when parameterized by treewidth, but they are W[1]-hard when parameterized by clique-width. To handle dense instances of problems that are hard when parameterized by clique-width, we have considered the notions of co-degeneracy and co-treewidth of a graph.

We have proposed a framework based on Bondy-Chvátal closure for working with co-degeneracy. Using this framework we showed that LONGEST CYCLE, LONGEST PATH and PATH COVER are FPT when parameterized by co-degeneracy. Additionally, using a different approach, we showed that EDGE DOMINATING SET is also FPT when parameterized by co-degeneracy. Conversely, we remark that GRAPH COLORING is para-NP-hard regarding co-degeneracy parameterization while the complexity of MAXCUT is left open. On the other hand, both GRAPH COLORING and MAXCUT are FPT when parameterized by co-treewidth.

We also have shown that PRECOLORING EXTENSION is fixed-parameter tractable taking the co-treewidth as parameter, while it is known to be W[1]-hard when parameterized by treewidth (see [27]). The same holds for EQUITABLE COLORING. In contrast, CLIQUE COVER EXTENSION and EQUITABLE CLIQUE COVER are FPT when parameterized by treewidth and W[1]-hard when parameterized by co-treewidth.

These results, which are summarized in Table 1, give evidence that co-degeneracy and co-treewidth are handy width parameters for dealing with problems for which FPT algorithms parameterized by clique-width are unlikely to exist.

Table 1 Parameterized complexity concerning treewidth, co-treewidth, co-degeneracy, and clique-width of graph problems addressed in this work. Courcelle and Olariu [20] proved that for every graph G we have $cwd(\overline{G}) \leq 2.cwd(G)$, thus the W[1]-hardness of CLIQUE COVER is implied from GRAPH COLORING. Also, the indicated para-NP-hardness are inherited from GRAPH COLORING or CLIQUE COVER. The main results presented in this work are highlighted in red.

	tw	*co-tw*	*co-deg*	*cw*
LONGEST PATH	FPT	FPT	FPT	W[1]-h
LONGEST CYCLE	FPT	FPT	FPT	W[1]-h
EDGE DOMINATING SET	FPT	FPT	FPT	W[1]-h
MAXIMUM CUT	FPT	FPT	open	W[1]-h
GRAPH COLORING	FPT	FPT	para-NP-h	W[1]-h
CLIQUE COVER	FPT	FPT	para-NP-h	W[1]-h
PRECOLORING EXTENSION	W[1]-h	FPT	para-NP-h	W[1]-h
EQUITABLE COLORING	W[1]-h	FPT	para-NP-h	W[1]-h
CLIQUE COVER EXTENSION	FPT	W[1]-h	para-NP-h	W[1]-h
EQUITABLE CLIQUE COVER	FPT	W[1]-h	para-NP-h	W[1]-h

We remark that min{treewidth,co-treewidth} seems to be a nice parameter between treewidth and clique-width. Note that every problem which can be expressed in both LinEMSO_2 and $\text{LinEMSO}_{\overline{2}}$ is solvable in FPT-time when parameterized by min{treewidth,co-treewidth}. Therefore, co-treewidth is a powerful tool to manipulate dense graphs.

We left the complexity of MAXCUT parameterized by co-degeneracy as an open problem. We remark that determining the complexity of MAXCUT seems to be a challenge even for co-planar graphs. Also, investigating the applicability of co-treewidth for problems that are hard when parameterized by treewidth is an interesting research direction. In particular, the complexity of LIST COLORING parameterized by co-treewidth is another interesting question.

Finally, we note that one can also consider parameters between co-degeneracy and co-treewidth such as co-contraction degeneracy, which is defined as the maximum degeneracy of a minor of the complement of G.

References

1 Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2):308–340, 1991.

2 Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM (JACM)*, 9(1):61–63, 1962.

3 Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1):1–45, 1998.

4 Hans L Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. In *International Colloquium on Automata, Languages, and Programming*, pages 196–207. Springer, 2013.

5 Hans L Bodlaender, Pål Grǿnås Drange, Markus S Dregi, Fedor V Fomin, Daniel Lokshtanov, and Michał Pilipczuk. A cˆkn 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016.

6 Hans L. Bodlaender and Klaus Jansen. On the complexity of the maximum cut problem. *Nordic Journal of Computing*, 7(1):14–31, 2000.

7 Hans L Bodlaender, Erik Jan Van Leeuwen, Johan MM Van Rooij, and Martin Vatshelle. Faster algorithms on branch and clique decompositions. In *International Symposium on Mathematical Foundations of Computer Science*, pages 174–185. Springer, 2010.

8 J Adrian Bondy and Vasek Chvátal. A method in graph theory. *Discrete Mathematics*, 15(2):111–135, 1976.

9 Richard B Borie, R Gary Parker, and Craig A Tovey. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica*, 7(1-6):555–581, 1992.

10 Andreas Brandstädt, Feodor F Dragan, Hoàng-Oanh Le, and Raffaele Mosca. New graph classes of bounded clique-width. *Theory of Computing Systems*, 38(5):623–645, 2005.

11 Stephan Brandt and Henk Jan Veldman. Degree sums for edges and cycle lengths in graphs. *Journal of graph theory*, 25(4):253–256, 1997.

12 Hajo Broersma, Zdeněk Ryjáček, and Ingo Schiermeyer. Closure concepts: a survey. *Graphs and Combinatorics*, 16(1):17–48, 2000.

13 Derek G Corneil and Udi Rotics. On the relationship between clique-width and treewidth. *SIAM Journal on Computing*, 34(4):825–847, 2005.

14 Stavros S Cosmadakis and Christos H Papadimitriou. The traveling salesman problem with many visits to few cities. *SIAM Journal on Computing*, 13(1):99–108, 1984.

15 B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000.

16 Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

17 Bruno Courcelle. The monadic second-order logic of graphs III: Tree-decompositions, minors and complexity issues. *RAIRO-Theoretical Informatics and Applications-Informatique Théorique et Applications*, 26(3):257–286, 1992.

18 Bruno Courcelle. The monadic second order logic of graphs VI: On several representations of graphs by relational structures. *Discrete Applied Mathematics*, 54(2-3):117–149, 1994.

19 Bruno Courcelle and Joost Engelfriet. *Graph structure and monadic second-order logic: a language-theoretic approach*, volume 138. Cambridge University Press, 2012.

20 Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1–3):77–114, 2000.

21 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

22 David P Dailey. Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete. *Discrete Mathematics*, 30(3):289–293, 1980.

23 Guilherme de C. M. Gomes, Carlos V. G. C. Lima, and Vinícius Fernandes dos Santos. Parameterized complexity of equitable coloring. *Discrete Mathematics & Theoretical Computer Science*, 21(1), 2019. URL: http://dmtcs.episciences.org/5464.

24 Rodney G. Downey and Michael R. Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013.

25 Pavel Dvořák, Dusan Knop, and Tomás Masarík. Anti-path cover on sparse graph classes. In Jan Bouda, Lukás Holík, Jan Kofron, Jan Strejcek, and Adam Rambousek, editors, *Proceedings 11th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science, MEMICS 2016, Telč, Czech Republic, 21st-23rd October 2016*, volume 233 of *EPTCS*, pages 82–86, 2016.

26 Eduard Eiben, Robert Ganian, Thekla Hamm, and O joung Kwon. Measuring what Matters: A Hybrid Approach to Dynamic Programming with Treewidth. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, volume 138 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42:1–42:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.MFCS.2019.42.

27 Michael R. Fellows, Fedor V. Fomin, Daniel Lokshtanov, Frances Rosamond, Saket Saurabh, Stefan Szeider, and Carsten Thomassen. On the complexity of some colorful problems parameterized by treewidth. *Information and Computation*, 209(2):143–153, 2011.

28 Fedor V Fomin, Petr A Golovach, Daniel Lokshtanov, and Saket Saurabh. Clique-width: on the price of generality. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 825–834. SIAM, 2009.

29 Fedor V Fomin, Petr A Golovach, Daniel Lokshtanov, and Saket Saurabh. Algorithmic lower bounds for problems parameterized by clique-width. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 493–502. SIAM, 2010.

30 Fedor V Fomin, Petr A Golovach, Daniel Lokshtanov, and Saket Saurabh. Intractability of clique-width parameterizations. *SIAM Journal on Computing*, 39(5):1941–1956, 2010.

31 Fedor V Fomin, Petr A Golovach, Daniel Lokshtanov, and Saket Saurabh. Almost optimal lower bounds for problems parameterized by clique-width. *SIAM Journal on Computing*, 43(5):1541–1563, 2014.

32 Jakub Gajarskỳ, Michael Lampis, and Sebastian Ordyniak. Parameterized algorithms for modular-width. In *International Symposium on Parameterized and Exact Computation*, pages 163–176. Springer, 2013.

33 Robert Ganian. Twin-cover: Beyond vertex cover in parameterized algorithmics. In *International Symposium on Parameterized and Exact Computation*, pages 259–271. Springer, 2011.

34 Robert Ganian, Petr Hliněnỳ, Jaroslav Nešetřil, Jan Obdržálek, Patrice Ossona de Mendez, and Reshma Ramadurai. When trees grow low: shrubs and fast MSO 1. In *International Symposium on Mathematical Foundations of Computer Science*, pages 419–430. Springer, 2012.

35 Martin Charles Golumbic and Udi Rotics. On the clique-width of some perfect graph classes. *International Journal of Foundations of Computer Science*, 11(03):423–443, 2000.

36 Michael Held and Richard M Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics*, 10(1):196–210, 1962.

37 Petr Hliněnỳ, Sang-il Oum, Detlef Seese, and Georg Gottlob. Width parameters beyond tree-width and their applications. *The computer journal*, 51(3):326–362, 2008.

38 Dušan Knop, Martin Koutecký, Tomáš Masařík, and Tomáš Toufar. Simplified algorithmic metatheorems beyond MSO: treewidth and neighborhood diversity. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 344–357. Springer, 2017.

39 Daniel Král', Jan Kratochvíl, Zsolt Tuza, and Gerhard J. Woeginger. Complexity of coloring graphs without forbidden induced subgraphs. In Andreas Brandstädt and Van Bang Le, editors, *Graph-Theoretic Concepts in Computer Science*, pages 254–262, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

40 Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012.

41 Clemens Lautemann. Logical definability of NP-optimisation problems with monadic auxiliary predicates. In *International Workshop on Computer Science Logic*, pages 327–339. Springer, 1992.

42 Meena Mahajan and Venkatesh Raman. Parameterizing above guaranteed values: Maxsat and maxcut. *J. Algorithms*, 31(2):335–354, 1999.

43 Sang-il Oum and Paul Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006.

44 Elena Prieto. The method of extremal structure on the k-maximum cut problem. In *Proceedings of the 2005 Australasian Symposium on Theory of Computing-Volume 41*, pages 119–126. Australian Computer Society, Inc., 2005.

45 Neil Robertson and Paul D. Seymour. Graph minors. X. obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991.

46 Sigve Hortemo Sæther and Jan Arne Telle. Between treewidth and clique-width. *Algorithmica*, 75(1):218–253, 2016.

47 Johan M. M. van Rooij, Hans L. Bodlaender, Erik Jan van Leeuwen, Peter Rossmanith, and Martin Vatshelle. Fast dynamic programming on graph decompositions, 2018. `arXiv:1806.01667`.

48 Johan MM Van Rooij, Hans L Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In *European Symposium on Algorithms*, pages 566–577. Springer, 2009.

49 Mihalis Yannakakis and Fanica Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38(3):364–372, 1980.