Clique-width: On the Price of Generality

Fedor V. Fomin Petr A. Golovach Daniel Lokshtanov Saket Saurabh

Department of Informatics, University of Bergen, N-5020 Bergen, Norway

{fedor.fomin|petr.golovach|daniello|saket.saurabh}@ii.uib.no

Abstract

Many hard problems can be solved efficiently when the input is restricted to graphs of bounded treewidth. By the celebrated result of Courcelle, every decision problem expressible in monadic second order logic is fixed parameter tractable when parameterized by the treewidth of the input graph. Moreover, for every fixed $k \geq 0$, such problems can be solved in linear time on graphs of treewidth at most k. In particular, this implies that basic problems like Dominating Set, Graph Coloring, Clique, and Hamiltonian Cycle are solvable in linear time on graphs of bounded treewidth.

A significant amount of research in graph algorithms has been devoted to extending this result to larger classes of graphs. It was shown that some of the algorithmic meta-theorems for treewidth can be carried over to graphs of bounded clique-width. Courcelle, Makowsky, and Rotics proved that the analogue of Courcelle's result holds for graphs of bounded clique-width when the logical formulas do not use edge set quantifications. Despite of its generality, this does not resolve the parameterized complexity of many basic problems concerning edge subsets (like Edge Dominating Set), vertex partitioning (like Graph Coloring), or global connectivity (like Hamiltonian Cycle). There are various algorithms solving some of these problems in polynomial time on graphs of clique-width at most k. However, these are not fixed parameter tractable algorithms and have typical running times $O(n^{f(k)})$, where n is the input length and f is some function.

It was an open problem, explicitly mentioned in several papers, whether any of these problems is fixed parameter tractable when parameterized by the clique-width, i.e. solvable in time $O(g(k) \cdot n^c)$, for some function g and a constant c not depending on k. In this paper we resolve this problem by showing that Edge Dominating Set, Hamiltonian Cycle, and Graph Coloring are $W[1]$-hard parameterized by clique-width. This shows that the running time $O(n^{f(k)})$ of many clique-width based algorithms is essentially the best we can hope for (up to a widely believed assumption from parameterized complexity, namely $FPT \neq W[1]$)—the price we pay for generality.

1 Introduction

One of the most frequent approaches for solving graph problems is based on decomposition methods. Tree decomposition, and the corresponding parameter, the treewidth of a graph, is one the most commonly used concepts. We refer to the surveys of Bodlaender [3] and Hlinený et al. [19] for further references on treewidth and related parameters. In the quest for alternate graph decompositions that can be applied to broader classes than graphs of bounded treewidth and still enjoy good algorithmic properties, Courcelle and Olariu [9] introduced the clique-width of a graph. Clique-width can be seen as a generalization of treewidth, in a sense that graphs of bounded treewidth also have bounded clique-width [5].

In recent years, clique-width has received much attention. Corneil, Habib, Lanlignel, Reed, and Rotics [4] show that graphs of clique-width at most 3 can be recognized in polynomial time. Fellows, Rosamond, Rotics, and Szeider [15] settled a long standing open problem by showing that computing clique-width is NP-hard. Oum and Seymour [24] describe an algorithm that, for any fixed k, runs in time $O(|V(G)|^9 \log |V(G)|)$ and computes $(2^{3k+2} - 1)$-expressions for a graph G of clique-width at most k. Recently Oum [20] improved this result by providing an algorithm computing $(8^k - 1)$-expressions in time $O(|V(G)|^3)$. It is also worth to mention here the related graph parameters NLC-width introduced by Wanke [27] and rank-width introduced by Seymour and Oum [24].

By the seminal result of Courcelle [6] (see also [1]), every decision problem on graphs expressible in monadic second order logic is fixed parameter tractable when parameterized by the treewidth of the input graph. For problems expressible in monadic second order logic with logical formulas that do not use edge set quantifications (so-called MS_1-logic), it is possible to extend the meta theorem of Courcelle to graphs of bounded clique-width. As it was shown by Courcelle, Makowsky, and Rotics [7], all problems expressible in MS_1-logic are fixed parameter tractable when parameterized by the clique-width of a graph.

There are many problems expressible in monadic second order logic that cannot be expressed in MS_1-logic. The most natural, are perhaps, Graph Coloring, Hamiltonian Cycle, and Edge Dominating

SET. It is known that these problems can be solved in polynomial time on graphs of bounded clique-width and a significant amount of the literature is devoted to algorithms for these problems and their generalizations. Polynomial time algorithms for GRAPH COLORING and its different generalizations including computations of chromatic and Tutte polynomials of graphs of bounded clique-width are given in [17, 16, 18, 21, 22, 23, 25, 26]. Polynomial time algorithms for HAMILTONIAN CYCLE are given in [27, 12] (in terms of NLC-width, which is a notion related to clique-width). Algorithms for EDGE DOMINATING SET are given in [21, 22]. The running time of all these algorithms on an n-vertex graph of clique-width at most k is $O(n^{f(k)})$, where f is some function of k. Since these problems are solvable in time $O(g(k) \cdot n^c)$, when the treewidth of the graph is at most k, the most natural question to ask is whether a similar behavior can be expected on graphs of bounded clique-width. The question on the existence of fixed parameter tractable algorithms (with clique-width being the parameter) for all these problems (or their generalizations) was asked by Gerber and Kobler [16], Kobler and Rotics [21, 22], Makowsky, Rotics, Averbouch, Kotek, and Godlin [23, 18].

Our results. In this paper we show that EDGE DOMINATING SET, HAMILTONIAN CYCLE, and GRAPH COLORING are $W[1]$-hard parameterized by clique-width, even when the expression tree is given. This resolves open questions raised in [16, 18, 21, 22, 23]. These are the first results distinguishing between treewidth and clique-width parameterizations.

2 Definitions and Preliminary results

We only consider finite undirected graphs without loops or multiple edges. The vertex set of a graph G is denoted by $V(G)$ and its edge set by $E(G)$. For $v \in V(G)$, by $E(v)$ we mean the set of edges incident to v. We denote by $\mathbf{tw}(G)$ the treewidth of the graph.

Parameterized Complexity: Parameterized complexity is a two dimensional framework for studying the computational complexity of a problem. (We refer to the book of Downey and Fellows [11] for an introduction to parameterized complexity.) One dimension is the input size n and another one is a *parameter* k. A problem is called *fixed parameter tractable* (FPT) if it can be solved in time $f(k) \cdot n^c$, where f is a computable function only depending on k and c is some constant. The basic complexity class for fixed parameter intractability is $W[1]$. To show that a problem is $W[1]$-hard, one needs to exhibit a parameterized reduction from a known $W[1]$-hard problems. Now we define the notion of parameterized reduction.

DEFINITION 2.1. *Let A, B be parameterized problems. We say that A is (uniformly many:1) reducible to B if there is an algorithm Φ which transforms (x, k) into $(x', g(k))$ in time $f(k)|x|^\alpha$, where $f, g : N \to N$ are arbitrary functions and α is a constant independent of k, so that $(x, k) \in A$ if and only if $(x', g(k)) \in B$.*

Clique-width: Let G be a graph, and k be a positive integer. A k-graph is a graph whose vertices are labeled by integers from $\{1, 2, \ldots, k\}$. We call the k-graph consisting of exactly one vertex labeled by some integer from $\{1, 2, \ldots, k\}$ an initial k-graph. The *cliquewidth* $\mathbf{cwd}(G)$ is the smallest integer k such that G can be constructed by means of repeated application of the following four operations: (1) *introduce*: construction of an initial k-graph labeled by i (denoted by $i(v)$), (2) *disjoint union* (denoted by $\oplus$), (3) *relabel*: changing all labels i to j (denoted by $\rho_{i \to j}$) and (4) *join*: connecting all vertices labeled by i with all vertices labeled by j by edges (denoted by $\eta_{i,j}$).

An *e*xpression tree of a graph G is a rooted tree T of the following form:

- The nodes of T are of four types i, $\oplus$, η and ρ.
- Introduce nodes $i(v)$ are leaves of T, corresponding to initial k-graphs with vertices v, which are labeled i.
- A union node $\oplus$ stands for a disjoint union of graphs associated with its children.
- A relabel node $\rho_{i \to j}$ has one child and is associated with the k-graph, which is the result of relabeling operation for the graph corresponding to the child.
- A join node $\eta_{i,j}$ has one child and is associated with the k-graph, which is the result of join operation for the graph corresponding to the child.
- The graph G is isomorphic to the graph associated with the root of T (with all labels removed).

The *w*idth of the tree T is the number of different labels appearing in T. If a graph G has $\mathbf{cwd}(G) \leq k$ then it is possible to construct a rooted expression tree T with *width* k of G

A well-known fact is that if the treewidth of a graph is bounded then its cliquewidth also is bounded. On the other hand, complete graphs have clique-width 2 and unbounded treewidth.

THEOREM 2.1. ([5]) *If graph G has treewidth at most t then $\mathbf{cwd}(G)$ is at most $k = 3 \cdot 2^{t-1}$. Moreover, an expression tree for G of width at most k can be constructed* *in FPT time (with treewidth being the parameter) from the tree decomposition of G.*

The second claim in Theorem 2.1 is not given explicitly in [5]. However it can be shown since the upper bound proof in [5] is constructive (see also [8, 13]). Note that if a graph has bounded treewidth then the corresponding tree decomposition can be constructed in linear time [2].

3 Graph Coloring — Chromatic Number

In this section, we prove that Graph Coloring is $W[1]$-hard parameterized by clique-width.

> Graph Coloring (or Chromatic Number): The chromatic number of a graph $G = (V(G), E(G))$ is the smallest number of colors $\chi(G)$ needed to color the vertices of G so that no two adjacent vertices share the same color.

Our reduction is from the Equitable Coloring problem parameterized by the number r of colors used, and the treewidth of the input graph. In the Equitable Coloring problem one is given a graph G and integer r and asked whether G can be properly r-colored in such a way that the number of vertices in any two color classes differs by at most 1. Notice that if n is divisible by r this implies that all color classes must contain the same number of vertices. In our reduction we will assume that in the instance we reduce from, n is divisible by r. For a justification of this assumption, if r does not divide n we can add a clique of size $n + r - \lfloor \frac{n}{r} \rfloor r$ to G. We reduce from the exact version of Equitable Coloring, that is, the version where we are looking for an equitable coloring of G with exactly r colors.

Theorem 3.1. ([14]) Equitable Coloring *is* $W[1]$*-hard parameterized by the treewidth t of the input graph and the number of colors r.*

Construction: On input (G, r) to Equitable Coloring, we construct an instance (G', r') of Graph Coloring as follows. We start with a copy of G and let $r' = r + nr$. We now add a clique P of size r' to G'. The clique P will function as a *palette* in our reduction, as we have to use all r' available colors to properly color it. We partition P into $r + 1$ parts as follows, $P = P^M \cup P_1 \cup P_2 \cdots \cup P_r$ where P^M has size r and P_i has size n for every i. We call P^M the main palette, and denote the vertices in P^M by p_i for $1 \le i \le r$. We add edges between every vertex of $P \setminus P^M$ and every vertex of the copy of G. For each vertex $u \in V(G)$ we assign a vertex $u_{P_i} \in P_i$ for every i. Now, for every $1 \le i \le r$ we add a set S_i of vertices. For each vertex $u \in V(G)$ we make a vertex u_{S_i} in S_i for every $1 \le i \le r$, and make u_{S_i} adjacent to u and the entire palette P *except for* u_{P_i} and p_i. We conclude the construction by adding a clique C_i of $n\frac{r-1}{r}$ vertices and making every vertex of C_i adjacent to all of the vertices of S_i and the entire palette except for P_i. See Figure 1 for an illustration.

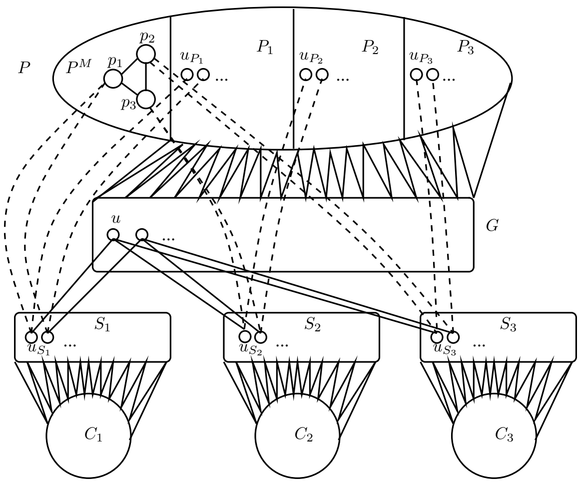

Figure 1: The figure shows the construction of G' for $r = 3$. The edges between vertices of S_i and P and between C_i and P are not shown. The dotted lines indicate non-edges.

Lemma 3.1. *If G has an equitable r-coloring ψ then G' has an r'-coloring ϕ.*

Proof. We construct a coloring ϕ of G' as follows. The coloring ϕ colors the copy of G in G' in the same way that ψ colors G. We color the palette, assigning a unique color to each vertex and making sure that the main palette P^M is colored using the same colors that are used to color the vertices of G. For every vertex u_{S_i} we color u_{S_i} with $\phi(p_i)$ if $\phi(u) \neq \phi(p_i)$ and with $\phi(u_{P_i})$ if $\phi(u) = \phi(p_i)$. We color every vertex of C_i with some color from P_i (a color used to color a vertex of P_i). To do this we need $n\frac{r-1}{r}$ different colors from P_i. Since exactly n/r vertices of G are colored with $\phi(p_i)$, exactly $n\frac{r-1}{r}$ of S_i are colored with $\phi(p_i)$ and thus n/r vertices of S_i are colored with colors of P_i. Hence there are $n\frac{r-1}{r}$ colors of P_i available to color C_i. Thus, ϕ is a proper r'-coloring of G concluding the proof. □

Lemma 3.2. *If G' has an r'-coloring ϕ then G has an equitable r-coloring ψ.*

Proof. We prove that the restriction of ϕ to the copy of G in G' in fact is an equitable r-coloring of G. Since ϕ can only use the colors of P^M, ϕ is a proper r-coloring of

G. It remains to prove that for any i between 1 and r, at most n/r vertices of G are colored with $\phi(p_i)$. Suppose for contradiction that there is an i such that more than n/r vertices of G are colored with $\phi(p_i)$. Then there are more than n/r vertices of S_i that are colored with colors of P_i. Since each such vertex must take a different color from P_i, there are less than $n\frac{r-1}{r}$ different colors of P_i available to color the vertices of C_i. However, since C_i is a clique on $n\frac{r-1}{r}$ vertices that must be colored with colors of P_i, this is a contradiction. ☐

Lemma 3.3. *If the treewidth of G is t, then the cliquewidth of G' is at most $k = 3 \cdot 2^{t-1} + 7r + 3$. Furthermore, an expression tree of width k for G' can be computed in FPT time.*

Proof. By Theorem 2.1 we can compute an expression tree for G of width at most $3 \cdot 2^{t-1}$ in FPT time. Our strategy is as follows. We first show how to modify the expression tree to give a width k expression tree for $G' \setminus (P^M \bigcup_{i=1}^r C_i)$. Then we change this tree into an expression tree for G'. In order to give an expression tree for G' we introduce the following extra labels.

- For every $1 \le i \le r$ the labels α_i, α_i^L and α_i^R for vertices in P_i.
- For every $1 \le i \le r$ the labels β_i, β_i^L and β_i^R for vertices in S_i.
- For every $1 \le i \le r$ the label ζ_i for vertices in C_i.
- A "work" label – γ^W, and a label γ^M for P_M.

In the expression tree for G, we replace every introduce-node $i(v)$ with a small expression tree $T_i(v)$. In $T_i(v)$, the vertex v is introduced with label γ^W and the vertices $v_{P_1}, \ldots, v_{P_r}$ and $v_{S_1}, \ldots, v_{S_r}$ are introduced with labels $\alpha_1, \ldots, \alpha_r$ and $\beta_1, \ldots, \beta_r$ respectively. Also, γ^W is joined to $\beta_1, \ldots, \beta_r$ and for every p, β_p is joined with every label in $\{\alpha_q : q \neq p\}$. Also, for every $p \neq q$, α_p is joined with α_q. Finally, γ^W is relabelled to i.

Now, for every union node in the expression tree (not the union nodes inside the T_i's) we add extra vertices on the edges incident to this node. On the edge from the node to its left child, we add nodes that relabel α_p to α_p^L and β_p to β_p^L for every p. Similarly, on the edge from the union node to its right child, we add nodes that relabel α_p to α_p^R and β_p to β_p^R for every p. Finally, on the edge from the union node to its parent we add nodes that first join every α_p^L with every β_q^R and α_q^R, join every α_p^R with every β_q^L, and then relabel every α_p^L and α_p^R to α_p and every β_p^L and β_p^R to β_p.

To conclude the construction of $G' \setminus (P^M \bigcup_{i=1}^r C_i)$ we need to add some extra nodes above the root of the expression tree. We add the edges between $P \setminus P^M$ and G by joining every α_p with all labels used for constructing G.

We now need to add the construction of P^M and $\bigcup_{i=1}^r C_i$ to our expression tree. We start by making C_p for every p between 1 and r. For every p we add a clique on $n\frac{r-1}{r}$ vertices labelled ζ_p. Every ζ_p is joined to β_p and for every pair $p \neq q$, ζ_p is joined with α_q.

Finally, we add the construction of P^M. For every i, we introduce the vertex p_i with label γ^W, join γ^W to α_j and ζ_j for every j, γ^W with β_j for every $j \neq i$ and finally join γ^W to γ^M and relabel γ^W to γ^M. This concludes the construction of G'. Notice that this expression tree for G' uses $k = 3 \cdot 2^{t-1} + 9r + 3$ labels. ☐

Lemmas 3.1, 3.2 and 3.3 together imply the following result.

Theorem 3.2. *The* Graph Coloring *problem is $W[1]$-hard when parameterized by clique-width. Moreover, this problem remains $W[1]$-hard even if the expression tree is given.*

4 Edge Dominating Set

In this section, we show that Edge Dominating Set probelm defined as below is $W[1]$-hard parameterized by clique-width.

> Edge Dominating Set: Given a graph $G = (V, E)$, find a minimum set of edges $X \subseteq E(G)$ such that every edge of G is either included in X or it is adjacent to at least one edge of X. The set X is called an *edge dominating set* of G.

Our reduction is from a variant of Capacitated Dominating Set problem.

4.1 Exact Saturated Capacitated Dominating Set: A *capacitated graph* is a pair (G, c) where G is a graph and $c \colon V(G) \to \mathbb{N}$ is a *capacity* function such that $1 \le c(v) \le \deg(v)$ for every vertex $v \in V(G)$ (sometimes we simply say that G is a capacitated graph if the capacity function is clear from the context). A set $S \subseteq V(G)$ is called a *capacitated dominating set* if there is a *domination mapping* $f \colon V(G) \setminus S \to S$ which maps every vertex in $(V(G) \setminus S)$ to one of its neighbors such that the total number of vertices mapped by f to any vertex $v \in S$ does not exceed its capacity $c(v)$. We say that for a vertex $u \in S$, vertices in the set $f^{-1}(u)$ are *dominated by u*. The Capacitated Dominating Set problem is formulated as follows: given a capacitated graph (G, c) and a positive integer k, determine whether there exists a capacitated dominating set S for G containing at most

k vertices. It was proved by Dom et al. [10] that this problem is $W[1]$-hard when parameterized by treewidth.

THEOREM 4.1. ([10]) CAPACITATED DOMINATING SET *is $W[1]$-hard parameterized by the treewidth t of the input graph and the solution size k.*

For the intractability proof of EDGE DOMINATING SET, we need a special variant of CAPACITATED DOMINATING SET problem which we call EXACT SATURATED CAPACITATED DOMINATING SET. Given a capacitated dominating set S, a $v \in S$ is called *saturated* if the corresponding domination mapping f maps $c(v)$ vertices to v, that is, $|f^{-1}(v)| = c(v)$. A capacitated dominating set $S \subseteq V(G)$ is called *saturated* if there is a domination mapping f which saturates all vertices of S. In EXACT SATURATED CAPACITATED DOMINATING SET a capacitated graph (G, c) and a positive integer k is given. The question is whether G has a saturated capacitated dominating set S with exactly k vertices.

LEMMA 4.1. *The* EXACT SATURATED CAPACITATED DOMINATING SET *problem is $W[1]$-hard when parameterized by clique-width. Moreover, this problem remains $W[1]$-hard even if the expression tree is given.*

Proof. We reduce from a exact version of the CAPACITATED DOMINATING SET problem parameterized by the treewidth of the input graph. In the exact version of the problem, the question is to determine whether there exists a capaciated dominating set of size exactly k. From the $W[1]$-hardness of CAPACITATED DOMINATING SET, it easily follows that even the exact version remains $W[1]$-hard for graphs of bounded treewidth.

Let r be a positive integer and $H_r(u)$ denote a capacitated graph rooted at vertex u. The graph $H_r(u)$ is constructed as follows. Its vertex set is given by $\{u, v, x_1, \dots, x_r, y_1, \dots, y_r\}$ and the edges are given by making u adjacent to all vertices x_i, making v adjacent to all vertices y_i, and finally adding edges $x_i y_j$, $1 \le i, j \le r$. We define the capacity function as follows: $c(v) = r - 1$, $c(x_i) = r + 1$ and $c(y_i) = i$ for all $i \in \{1, 2, \dots, r\}$ (note that the capacity function is not defined for the root u).

Let (G, c) be a capacitated graph, $u \in V(G)$, and $r \ge \max\{3, c(u)+1\}$. We add a copy of $H_r(u)$ to G with u being its root. Let G' be the resulting capacitated graph. We now prove two auxiliary claims about the graph G'.

CLAIM 1. *Any capacitated dominating set S with the domination mapping f in G can be extended to the capacitated dominating set in G' in such a way that all vertices of $H_r(u)$ are saturated.*

Proof. Let S be a capacitated dominating set in G with the domination mapping f. We define s to be $|f^{-1}(u)|$ if $u \in S$ and $c(u)$ otherwise. Let $S' = S \cup \{v, y_j\}$ where $j = r - c(u) + s$. The mapping f is extended as follows: $f(x_i) = u$ for $1 \le i \le c(u) - s$, $f(x_i) = y_j$ for $i > c(u) - s$, and $f(y_i) = v$ for all $i \ne j$. It can be easily seen that this is the claimed extention. ☐

CLAIM 2. *Every saturated capacitated dominating set in G' contains exactly two vertices from $V(H_r(u)) \setminus \{u\}$.*

Proof. Let S' be a saturated capacitated dominating set in G' and f be its corresponding domination mapping. We first show that S' does not contain any x_i's. Suppose that some vertex x_i is included in S'. Then because of capacity constraint that $c(x_i) = r + 1$, it implies that $y_1, y_2, \dots, y_r \notin S'$ and $f(y_j) = x_i$ for all these vertices. Therefore $v \in S'$ but clearly this vertex can not be saturated. Hence, $x_1, x_2, \dots, x_r \notin S'$. Now we show that v must be in S'. Assume to the contrary that $v \notin S'$. Then $y_1, y_2, \dots, y_r \in S'$, as they need to be dominated. But these vertices can not be saturated since $\sum_{i=1}^{r} c(y_i) = 1 + \cdots + r = \frac{r(r+1)}{2} > r + 1$. This means that $v \in S'$. The capacity of v is $r - 1$, hence at most one vertex y_i can be included in S'. On the other hand since $c(u) < r$, there exists at least one vertex x_j such that $f(x_j) \ne u$. Hence to dominate this vertex we need a vertex $y_i \in S'$. This concludes the proof. ☐

Now we are ready to complete the proof of the lemma. Let (G, c) be a capacitated graph with the vertex set $\{u_1, u_2, \dots, u_n\}$, $r = \max\{c(v) : v \in V(G)\} + 2$. For every vertex u_i, we add a copy of $H_r(u_i)$ to G with u_i being its root. Let the resulting capacitated graph be denoted by H. By applying Claims 1 and 2 we conclude that G has a capacitated dominating set of the size k if and only if H has an exact saturated dominating set of the size $k + 2n$.

It remains to prove that if the treewidth of G is bounded then the clique-width of H is bounded. Let $\mathbf{tw}(G) \le t$. The by Theorem 2.1 $\mathbf{cwd}(G) \le 3 \cdot 2^{t-1}$ and it is possible to construct an expression tree of width at most $w = 3 \cdot 2^{t-1}$ in FPT time. We prove that $\mathbf{cwd}(H) \le w + 4$. Assume that the construction of the labeled graph G uses labels from the set $\{\alpha_1, \dots, \alpha_w\}$. To construct H from G we use additional labels $\{\beta_1, \beta_2, \beta_3, \beta_4\}$.

When a vertex u having a label α_j is introduced we do the following sequence of operations: $\alpha_j(u)$, $\beta_1(x_i)$ and $\beta_2(y_i)$ for all $i \in \{1, \dots, r\}$, and $\beta_3(v)$. After this we apply following operations: η_{α_j, β_1}, η_{β_1, β_2}, η_{β_2, β_3} and $\rho_{\beta_i \to \beta_4}$ for $i = 1, 2, 3$. We omit the union operations in this description: it is assumed that if some vertex is introduced then this operation is automatically performed. Join, union and relabel operations with labels $\{\alpha_1, \dots, \alpha_w\}$ are done as it is done for the expression tree of G. This concludes the construction of expression tree for H. $\square$

4.2 Intractability of Edge Dominating Set problem: In this section we show that Edge Dominating Set is $W[1]$-hard parameterized by clique-width by giving a reduction from Exact Saturated Dominating Set. We start with descriptions of auxiliary gadgets.

Construction: We start with descriptions of auxiliary gadgets. Let $s \leq t$ be positive integers. We construct a graph $F_{s,t}$ with the vertex set $\{x_1, \dots, x_s, y_1, \dots, y_s, z_1, \dots, z_t\}$ and edges $x_i y_i$, $1 \leq i \leq s$ and $y_i z_j$, $1 \leq i \leq s$ and $1 \leq j \leq t$. Basically we have complete bipartite graph between y_i's and z_j's with pendent vertices attached to y_i's. The vertices $z_1, z_2, \dots, z_t$ are called *roots* of $F_{s,t}$. This graph has the following property.

Lemma 4.2. *Any set of s edges incident to vertices $y_1, \dots, y_s$ forms an edge dominating set in $F_{s,t}$. Furthermore, let G be a graph obtained by the union of $F_{s,t}$ with some other graph H such that $V(F_{s,t}) \cap V(H) = \{z_1, \dots, z_t\}$. Then every edge dominating set of G contains at least s edges from $F_{s,t}$.*

The proof of the lemma follows from the fact that every edge dominating set includes at least one edge from $E(y_i)$ for $i \in \{1, \dots, s\}$.

Now we describe our reduction. Let (G, c) be a capacitated graph with the vertex set $\{u_1, \dots, u_n\}$, and k be a positive integer. For every vertex u_i, the set U_i with $c(u_i)$ vertices is introduced, and then vertex sets $\{v_1, \dots, v_n\}$ and $\{w_1, \dots, w_n\}$ are added. For every edge $u_i u_j \in E(G)$, all vertices of U_i are joined with v_j and all vertices of U_j are joined with v_i by edges. Then every vertex v_i is joined to its counterpart w_i and to every vertex v_i we add one additional leaf (a pendent vertex). Now vertex sets $\{a_1, \dots, a_n\}$ and $\{b_1, \dots, b_n\}$ are constructed, and vertices a_i are made adjacent to all vertices of U_i, w_i and b_i. For every vertex b_i, a set R_i of $c(u_i) + 1$ vertices is added and b_i is made adjacent to all the vertices in R_i. Then we add to every vertex of $R_1 \cup R_2 \cup \dots \cup R_n$ a path of length two. Let X be the set of middle vertices of these paths. We denote the obtained graph by G' (see Fig 2). Finally, we introduce three copies of $F_{s,t}$:

- a copy of $F_{n-k,n}$ with roots $\{a_1, \dots, a_n\}$,
- a copy of $F_{k,n}$ with roots $\{b_1, \dots, b_n\}$, and a
- a copy of $F_{n,r}$ where $r = \sum_{i=1}^{n} c(u_i)$ with roots in X.

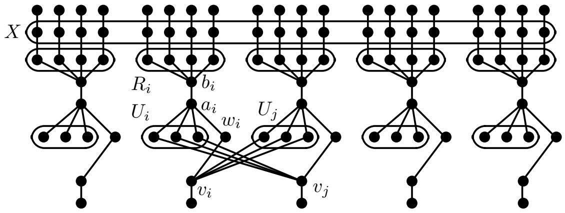

Figure 2: Graph G'

Let this final resulting graph be H.

Lemma 4.3. *A graph G has an exact saturated dominating set of the size k if and only if H has an edge dominating set of cardinality at most $2n + r$.*

Proof. Let S be an exact saturated dominating set of the size k in G and f be its corresponding domination mapping. For convenience (without loss of a generality) we assume that $S = \{u_1, \dots, u_k\}$. We construct the edge dominating set as follows. First we select an edge emanating from every vertex in the set $\{v_1, \dots, v_n\}$. For every vertex v_i, $1 \leq i \leq k$, the edge $v_i w_i$ is selected. Now let us assume that $k < i \leq n$ and $f(u_i) = u_j$. We choose a vertex u in U_j which is not incident to already chosen edges and add the edge uv_i to our set. Notice that we always have such a choice of $u \in U_j$ as $c(u_j) = |U_j|$. We observe that these edges already dominates all the edges in the sets $E(v_i)$, $1 \leq i \leq n$, and in sets $E(u)$ for $u \in U_1 \cup \dots \cup U_k \cup \{w_1, \dots, w_k\}$. Now we add $n - k$ edges from $F_{n-k,n}$ which are incident to vertices in $\{a_{k+1}, \dots, a_n\}$ and k edges from $F_{k,n}$ which are incident to $\{b_1, \dots, b_k\}$. Then $r - n$ matching edges joining vertices of $R_{k+1}, \dots, R_n$ to the vertices of X are included in the set. Finally, we add n edges form $F_{n,r}$ which are incident to vertices of X which are adjacent to vertices of $R_1, \dots, R_k$. Since S is an exact capacitated dominating set, $\sum_{i=1}^{k}(c(u_i) + 1) = n$, and from our description it is clear that the resulting set is an edge dominating set of size $2n + r$ for H.

We proceed to proving the other direction of the equivalence. Let L be an edge dominating set of cardinality at most $2n + r$. The set L is forced to contain at least one edge from every $E(v_i)$, at least $n - k$ edges from $F_{n-k,n}$, at least k edges from $F_{k,n}$, and at least one edge from $E(x)$ for all $x \in X$ because of pendent edges. This implies that $|L| = 2n + r$, and L contains exactly one edge from every $E(v_i)$, exactly $n - k$ edges from $F_{n-k,n}$, exactly k edges from $F_{k,n}$, and exactly one edge from $E(x)$ for all $x \in X$. Every edge $a_i b_i$ needs to be dominated by some edge of L, in particular it must be dominated from either an edge of $F_{n-k,n}$ or $F_{k,n}$. Let $I =$

$\{i : a_i \text{ is incident to an edge from } L \cap E(F_{n-k,n})\}$ and $J = \{j : b_j \text{ is incident to an edge from } L \cap E(F_{k,n})\}$. The above constraints on the set L implies that $|I| = n - k$, $|J| = k$, and these sets form a partition of $\{1, \dots, n\}$. The edges which join vertices b_i and R_i for $i \in I$ are not dominated by edges from $L \cap E(F_{k,n})$. Hence to dominate these edges we need at least $\sum_{i \in I} |R_i|$ edges which connect sets R_i and X. Since at least n edges of $F_{n,r}$ are included in L, we have that $\sum_{i \in I} |R_i| \leq r - n$ and $\sum_{j \in J} |R_j| = r - \sum_{i \in I} |R_i| \geq r - (r - n) \geq n$. Let $S = \{u_j : j \in J\}$. Clearly, $|S| = k$. Now we show that S is a saturated capacitated dominating set. For $j \in J$, edges which join a vertex a_j to U_j and w_j are not dominated by edges from $L \cap E(F_{n-k,k})$, and hence they have to be dominated by edges from sets $E(v_i)$. Since $n \leq \sum_{j \in J} |R_j| = \sum_{j \in J}(|U_j| + 1)$, there are exactly n such edges, and every such edge must be dominated by exactly one edge from L. An edge $a_j w_j$ can only be dominated by edge $v_j w_j$. We also know that $L \cap E(v_i) \neq \emptyset$ for all $i \in \{1, \dots, n\}$ and hence for every v_i, $i \notin J$, there is exactly one edge which joins it with some vertex $u \in U_j$ for some $j \in J$. Furthermore, all these edges are not adjacent, that is, they form a matching. We define $f(u_i) = u_j$ for $i \notin J$. From our construction it follows that f is a domination mapping for S and S is an exact saturated dominating set in G. □

The next lemma shows that if the graph G we started with has bounded clique-width then H also has bounded clique-width.

Lemma 4.4. *If* $\mathbf{cwd}(G) \leq t$ *then* $\mathbf{cwd}(H) \leq 2t + 16$, *and an expression tree for* H *of width at most* $w = 2t + 16$ *can be constructed in a polynomial time from the expression tree for* G.

Proof. The graph G is of clique-width at most t. Suppose that the expression tree for G uses t-labels $\{\alpha_1, \dots, \alpha_t\}$. To construct the expression tree for H we need following additional labels:

- Labels $\beta_1, \dots, \beta_t$ for the vertices in $U_1, \dots, U_n$.
- Labels ξ_1, ξ_2, and ξ_3 for attaching $F_{n-k,n}$, $F_{k,n}$ and $F_{n,r}$ respectively.
- Labels $\zeta_1, \dots, \zeta_4$ for marking some vertices like $w_1, \dots, w_n$.
- Working labels $\gamma_1, \dots, \gamma_9$.

When a vertex $u_i \in V(G)$ labeled α_j is introduced, we perform following set of operations. First we introduce following vertices with some working labels: v_i with label γ_1, $c(u_i)$ vertices of U_i with label γ_2, the vertex w_i with label γ_3, and the additional vertex (the leaf attached to v_i) with label γ_4. Now we join the vertex labelled with γ_1 to vertices labelled with γ_3 and γ_4 (basically joining v_i with w_i and its pendent leaf). Finally, we relabel γ_4 to ζ_1 and γ_1 to β_j. Now we introduce vertices a_i and b_i with labels γ_5 and γ_6 respectively. Then we join the vertex labelled γ_4 (a_i) with all the vertices labelled with γ_2, γ_3 and γ_6 (U_i, w_i, b_i). The join operation is followed by relabeling γ_3 to ζ_2, γ_2 to α_j and γ_5 with ξ_1.

Now we want to make the vertices of R_i and the paths attached to it. To do so we perform following operations $c(u_i) + 1$ times: (a) introduce three nodes labelled with γ_7, γ_8 and γ_9 (b) join γ_6 with γ_7, γ_7 with γ_8 and γ_8 with γ_9 and (c) finally we relabel γ_6 to ξ_2, γ_7 to ζ_3, γ_8 to ξ_3 and γ_9 to ζ_4. We omit the union operations from the description and assume that if some vertex is introduced then this operation is performed.

If in the expression tree of G, we have join operation between two labels say α_i and α_j then we simulate this by applying join operations between α_i and β_j and α_j and β_i. The relabel operation in the expression tree of G, that is, relabel α_i to α_j is replaced by relabel α_i to α_j and relabel β_i to β_j. Union operations in the expression tree is done as before.

Finally to complete the expression tree for H, we need to add $F_{n-k,n}$, $F_{k,n}$ and $F_{n,r}$. Notice that all the vertices in $\{a_1, \dots, a_n\}$, $\{b_1, \dots, b_n\}$ and X are labelled ξ_1, ξ_2 and ξ_3 respectively. From here we can easily add $F_{n-k,n}$, $F_{k,n}$ and $F_{n,r}$ with root vertices $\{a_1, \dots, a_n\}$, $\{b_1, \dots, b_n\}$ and X respectively by using working labels. This concludes the description for the expression tree for H. □

Lemmas 4.3 and 4.4 together imply the following result.

Theorem 4.2. *The* Edge Dominating Set *problem is* $W[1]$*-hard when parameterized by clique-width. Moreover, the problem remains* $W[1]$*-hard even if the expression tree is given.*

5 Hamiltonian Cycle Problem

In this section we show that the Hamiltonian Cycle problem is $W[1]$-hard for the graphs of bounded clique-width.

> Hamiltonian Cycle: Given a graph G, check whether there exists a cycle passing through every vertex of G.

Our reduction is from the Capacitated Dominating Set problem described in Section 4.1 and shown to be $W[1]$-hard in Theorem 4.1.

Construction: We start with descriptions of auxiliary gadgets. We denote by L_1, the graph with the vertex set $\{x, y, z, a, b, c, d\}$ and the edge set $\{xa, ab, bc, cd, dy, bz, cz\}$. Let P_1 be the path $xabzcdy$, and $P_2 = xabcdy$. (See Figure 3). We abstract a property of this graph in the following lemma.

Lemma 5.1. *Let G be a Hamiltonian graph which contains L_1 as an induced subgraph. Furthermore, if all the edges of G which are not edges of L_1 are only incident to the vertices x, y, and z, then every Hamiltonian cycle in G either includes the path P_1 or the path P_2 as a segment.*

Our second auxiliary gadget is the graph L_2. This graph has $\{x, y, z, s, t, a, b, c, d, e, f, g, h\}$ as its vertex set. We first include following $\{xa, ab, bz, cz, cd, dy, se, ef, fb, ch, hg, gt\}$ in its edge set. Then x, y-path of length ten $xw_1 \cdots w_9 y$ is added, and edges $fw_3, w_1w_6, w_4w_9, w_7h$ are included in the set of edges. Let $P = xabzcdy$, $R_1 = sefbaxw_1w_2 \dots w_9ydchgt$, and $R_2 = sefw_3w_2w_1w_6w_5w_4w_9w_8w_7hgt$. (See Figure 3.) This graph has the following property.

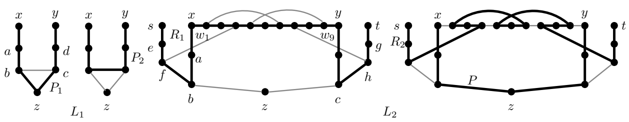

Figure 3: Graphs L_1 and L_2. Paths P_1, P_2, R_1, R_2 and P are shown by thick lines

Lemma 5.2. *Let G be a Hamiltonian graph which contains L_2 as an induced subgraph, and edges of G which are not edges of L_1 are incident only to the vertices x, y, z, s, t. Then every Hamiltonian cycle in G includes either the path R_1 or two paths P and R_2 as segments.*

The lemma follows from the presence of degree 2 vertices in the graph L_2

Now we are ready to describe our reduction. Let (G, c) be a capacitated graph with the vertex set $\{v_1, \dots, v_n\}$, m edges, and let k be a positive integer. For every vertex v_i, four vertices a_i, b_i, c_i and w_i are introduced and the vertices b_i and c_i are joined by $c(v_i) + 1$ paths of length two. Let C_i denote the set of middle vertices of these paths, and $X_i = C_i \cup \{a_i, b_i, c_i\}$. Then a copy L_2^i of the graph L_2 with $z = w_i$ is added and vertices x and y of this gadget are joined by edges to a_i and b_i respectively. By s_i and t_i we denote the vertices s and t of L_2^i. For every ordered pair $\{v_i, v_j\}$ such that $v_iv_j \in E(G)$, a copy L_2^{ij} of L_2 is attached with $z = w_j$ and vertices x and y made adjacent to all the vertices of C_i. The vertices corresponding to s and t are called s_{ij} and t_{ij} in L_2^{ij}. Furthermore, let x_{ij} and y_{ij} denote the vertices corresponding to x and y in L_2^{ij}. The path corresponding to P in L_2^i is called P^i. Similarly, the path corresponding to P, R_1 and R_2 are called P^{ij}, R_1^{ij} and R_2^{ij} respectively in L_2^{ij}. Denote the obtained graph by $G'(c)$.

In the next step we add two vertices g and h which are joined by $\sum_{i=1}^{n}(c(v_i)+4)+n+2m+1$ paths of length two. Let Y be the set of middle vertices of these paths. All vertices s_i, t_i, s_{ij} and t_{ij} are joined by edges with all vertices of Y. For every vertex r such that $r \in X_i$ (recall $X_i = C_i \cup \{a_i, b_i, c_i\}$), $i \in \{1, \dots, n\}$, a copy L_1^r of L_1 with $z = r$ is attached and the vertices x, y of this gadget are joined to all vertices of Y. We let x_r and y_r denote the vertices corresponding to x and y in L_1^r. Similarly P_1^r and P_2^r denotes paths in L_1^r corresponding to P_1 and P_2 respectively. Please refer to Figure 4 for an illustration.

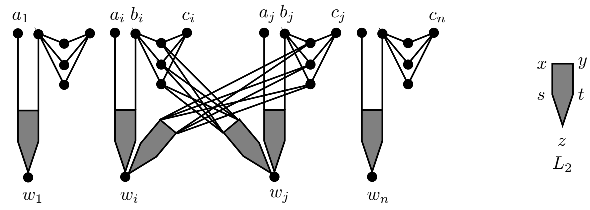

Figure 4: Graph $G'(c)$

Finally we add $k+1$ vertices, namely $\{p_1, \dots, p_{k+1}\}$, and make it adjacent to all the vertices $\{a_i, c_i : 1 \leq i \leq n\}$ and to g and h. Let this resultant graph be H. The construction of H can easily be done in time polynomial in n and m.

Lemma 5.3. *A graph (G, c) has a capacitated dominating set of size at most k if and only if H has a Hamiltonian cycle.*

Proof. Let S be a capacitated dominating set of size at most k in (G, c) with the corresponding dominating mapping f. Without loss of a generality we assume that $|S| = k$ and $S = \{v_1, \dots, v_k\}$. The Hamiltonian cycle we are trying to construct is naturally divided into $k + 1$ parts by the vertices $\{p_1, \dots, p_{k+1}\}$. We construct the Hamiltonian cycle starting from the vertex p_1. Assume that the part of the cycle up to the vertex p_i is already constructed. We show how to construct the part from p_i to p_{i+1}. We include the edge p_ia_i in it. We add to the cycle the path P^i and two edges, which join the endpoints of P_i with a_i and b_i. Let

$J = \{j \colon f(v_j) = v_i\}$. If $J = \emptyset$ then a $b_i - c_i$-path of length two which goes through one vertex of C_i is included in the cycle. Otherwise all paths P^{ij} for $j \in J$ are included in our cycle as follows. We consider the paths P^{ij} in the increasing order of indices in J and add them to our cycle. We take the first path say $P^{ij'}$ and attach $x_{ij'}$ and $y_{ij'}$ to a pair of vertices $\{j_1, j_2\}$ in C_i. Suppose iteratively we have included first $l \geq 1$ paths in J, and the l^{th} path is incident to some $\{j_l, j_{l+1}\}$ in C_i, now we attach the $(l+1)^{th}$ path by attaching x_{it} of this to j_{l+1} and y_{it} of this to j_{l+2}, where j_{l+2} is a new vertex not incident to any previously included paths. We can always find such a vertex as $|J| \leq c(v_i) = |C_i| - 1$. Now we include the edge $b_i j_1$ and $j_{|J|+1} c_i$. Finally we include the edge $c_i p_{i+1}$.

When the vertex p_{k+1} is reached we move to the set Y. Note that at this stage all vertices $\{w_1, \ldots, w_n\}$ are already included in our cycle. We start by including the edge $p_{k+1} g$. We will add following segments to our cycle an connect them appropriately.

- For every L_2^i we add the path R_1^i to the cycle if P^i was not included to it, and include the path R_2^i otherwise. The number of such paths is n.

- Similarly, for every L_2^{ij}, the path R_1^{ij} is added to the cycle if P^{ij} was not included to it, else the path R_2^{ij} is added. Note that $2m$ such paths are included to the cycle.

- For every vertex r such that $r \in X_i$ for some $i \in \{1, \ldots, n\}$, the path P_2^r is included in the cycle if r is already included in the constructed part of the cycle, else the path P_1^r is added. Clearly, we add $\sum_{i=1}^{n}(c(v_i) + 4)$ paths.

Finally the total number of paths we will add is $\sum_{i=1}^{n}(c(v_i) + 4) + n + 2m = |Y| - 1$. We add the segments of the paths mentioned with the help of vertices in Y, in the way we added the paths P^{ij} with the help of vertices in C_i. Let the end points of the resultant joined path be $\{q_1, q_2\}$. Notice that (a) $q_1, q_2 \in Y$ and (b) this path include all the vertices of Y. Now we add edges gq_1, $q_2 h$ and hp_1. This completes the construction of the Hamiltonian cycle.

For the reverse direction of the proof, we assume that we have been given C, a Hamiltonian cycle in H. Let $S = \big\{v_i \mid p_j a_i \in E(C), a_i p_s \notin E(C), j \neq s$, for some $j \in \{1, 2, \ldots, k+1\}\big\}$. We prove that S is a capacitated dominating set in G of cardinality at most k. We first argue about the size of S, clearly its size is upper bounded by $k+1$. To argue that it is at most k, it is enough to observe that by Lemmas 5.1 and 5.2 either $p_j g$ or $p_j h$ must be in $E(C)$ for some $j \in \{1, \ldots, k+1\}$. Now we show that S is indeed a capacitated dominating set. Our proof is based on following observations.

- Every vertex w_j, either appear in a vertex segment, that is, P^j or an edge segment, that is, P^{ij} for some $j \in \{1, \ldots, n\}$ in C.

- If some P^{ij} appear as a segment in C, then from the gadgets $L_1^{b_i}$ and $L_1^{c_i}$ the paths $P_2^{b_i}$ and $P_2^{c_i}$ are part of C. Hence the only way to include b_i in C is by using the edge incident to it from the gadget L_2^i. This implies that from the gadget L_2^i we use the path P^i and two edges, which join the endpoints of P_i with a_i and b_i.

- By Lemma 5.1 the cycle contains the edge which joins a_i to some vertex in $\{p_1, \ldots, p_{k+1}\}$.

Now given $v_j \in V(G) \setminus S$, for the domination function f, we assign it to v_i for which P^{ij} is segment in C. Clearly $v_i \in S$ as by above observation there exits a $j \in \{1, 2, \ldots, k+1\}$ such that $p_j a_i \in E(C)$, $a_i p_s \notin E(C)$ and $j \neq s$. For every $v_i \in S$, the set $f^{-1}(v_i)$ contains at most $c(v_i)$ vertices as $|C_i| = c(v_i) + 1$. This concludes the proof. □

The proof of the next lemma is similar in spirit to Lemmas 3.3 and 4.4 and has been omitted due to space restrictions.

LEMMA 5.4. *If* $\mathbf{tw}(G) \leq t$ *then* $\mathbf{cwd}(H) \leq 9 \cdot 2^{\max\{2t, 24\}} + 12$ *and an expression tree for* H *of width at most* $w = 9 \cdot 2^{\max\{2t, 24\}} + 12$ *can be constructed in FPT time.*

Lemmas 5.3 and 5.4 together imply the following result.

THEOREM 5.1. *The* HAMILTONIAN CYCLE *problem is* $W[1]$*-hard when parameterized by clique-width. Moreover, this problem remains* $W[1]$*-hard even if the expression tree is given.*

6 Conclusions

In this article, we settled the computational complexity of several important problems parameterized by the clique-width of the input graph. Our results show that the existing algorithms for EDGE DOMINATING SET, HAMILTONIAN CYCLE, and GRAPH COLORING in graphs of bounded clique-width essentially are the best one can hope for, unless an unlikely collapse in parameterized complexity occurs. The problems we prove $W[1]$ hard parameterized by clique-width are expressible in monadic second order logic and thus can be solved in linear time in graphs of bounded tree-width. Therefore our results illustrate the trade-off between expressive power and computational tractability. Finally, we leave the following as an open problem—what is the complexity of MAX CUT parameterized by the clique-width of the input graph?

Acknowledgement. The authors thank Mike Fellows for introducing us to the "dark art" of parameterized reductions.

References

[1] S. ARNBORG, J. LAGERGREN, AND D. SEESE, *Easy problems for tree-decomposable graphs*, J. Algorithms, 12 (1991), pp. 308–340.

[2] H. L. BODLAENDER, *A linear-time algorithm for finding tree-decompositions of small treewidth*, SIAM J. Comput., 25 (1996), pp. 1305–1317.

[3] ———, *A partial k-arboretum of graphs with bounded treewidth*, Theoret. Comput. Sci., 209 (1998), pp. 1–45.

[4] D. G. CORNEIL, M. HABIB, J.-M. LANLIGNEL, B. A. REED, AND U. ROTICS, *Polynomial time recognition of clique-width ≤ 3 graphs (extended abstract)*, in LATIN'00, vol. 1776 of LNCS, Springer, 2000, pp. 126–134.

[5] D. G. CORNEIL AND U. ROTICS, *On the relationship between clique-width and treewidth*, SIAM J. Comput., 34 (2005), pp. 825–847.

[6] B. COURCELLE, *The monadic second-order logic of graphs. III. Tree-decompositions, minors and complexity issues*, RAIRO Inform. Théor. Appl., 26 (1992), pp. 257–286.

[7] B. COURCELLE, J. A. MAKOWSKY, AND U. ROTICS, *Linear time solvable optimization problems on graphs of bounded clique-width*, Theory Comput. Syst., 33 (2000), pp. 125–150.

[8] ———, *On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic*, Discrete Appl. Math., 108 (2001), pp. 23–52. International Workshop on Graph-Theoretic Concepts in Computer Science (Smolenice Castle, 1998).

[9] B. COURCELLE AND S. OLARIU, *Upper bounds to the clique width of graphs*, Discrete Appl. Math., 101 (2000), pp. 77–114.

[10] M. DOM, D. LOKSHTANOV, S. SAURABH, AND Y. VILLANGER, *Capacitated domination and covering: A parameterized perspective*, in IWPEC'08, vol. 5018 of LNCS, Springer, 2008, pp. 78–90.

[11] R. G. DOWNEY AND M. R. FELLOWS, *Parameterized complexity*, Monographs in Computer Science, Springer-Verlag, New York, 1999.

[12] W. ESPELAGE, F. GURSKI, AND E. WANKE, *How to solve np-hard graph problems on clique-width bounded graphs in polynomial time*, in WG'01, vol. 2204 of LNCS, Springer, 2001, pp. 117–128.

[13] W. ESPELAGE, F. GURSKI, AND E. WANKE, *Deciding clique-width for graphs of bounded tree-width*, J. Graph Algorithms Appl., 7 (2003), pp. 141–180 (electronic).

[14] M. R. FELLOWS, F. V. FOMIN, D. LOKSHTANOV, F. A. ROSAMOND, S. SAURABH, S. SZEIDER, AND C. THOMASSEN, *On the complexity of some colorful problems parameterized by treewidth*, in COCOA, 2007, pp. 366–377.

[15] M. R. FELLOWS, F. A. ROSAMOND, U. ROTICS, AND S. SZEIDER, *Clique-width minimization is NP-hard (extended abstract)*, in STOC'06, ACM, 2006, pp. 354–362.

[16] M. U. GERBER AND D. KOBLER, *Algorithms for vertex-partitioning problems on graphs with fixed clique-width*, Theoret. Comput. Sci., 299 (2003), pp. 719–734.

[17] O. GIMÉNEZ, P. HLINENÝ, AND M. NOY, *Computing the tutte polynomial on graphs of bounded clique-width*, SIAM J. Discret. Math., 20 (2006).

[18] B. GODLIN, T. KOTEK, AND J. MAKOWSKY, *Evaluations of graph polynomials*, in WG'08, LNCS, Springer, 2008, p. to appear.

[19] P. HLINENÝ, S. IL OUM, D. SEESE, AND G. GOTTLOB, *Width parameters beyond tree-width and their applications*, Comput. J., 51 (2008), pp. 326–362.

[20] S. IL OUM, *Approximating rank-width and clique-width quickly*, ACM Trans. Algorithms, (2008), p. to appear.

[21] D. KOBLER AND U. ROTICS, *Polynomial algorithms for partitioning problems on graphs with fixed clique-width (extended abstract)*, in SODA'01, ACM-SIAM, 2001, pp. 468–476.

[22] ———, *Edge dominating set and colorings on graphs with fixed clique-width*, Discrete Appl. Math., 126 (2003), pp. 197–221.

[23] J. MAKOWSKY, U. ROTICS, I. AVERBOUCH, AND B. GODLIN, *Computing graph polynomials on graphs of bounded clique-width*, in WG'06, LNCS, Springer, 2006, pp. 191–204.

[24] S.-I. OUM AND P. SEYMOUR, *Approximating clique-width and branch-width*, J. Combin. Theory Ser. B, 96 (2006), pp. 514–528.

[25] M. RAO, *MSOL partitioning problems on graphs of bounded treewidth and clique-width*, Theoret. Comput. Sci., 377 (2007), pp. 260–267.

[26] K. SUCHAN AND I. TODINCA, *On powers of graphs of bounded NLC-width (clique-width)*, Discrete Appl. Math., 155 (2007), pp. 1885–1893.

[27] E. WANKE, *k-NLC graphs and polynomial algorithms*, Discrete Appl. Math., 54 (1994), pp. 251–266. Efficient algorithms and partial *k*-trees.