MULTISCALE MODEL. SIMUL.
Vol. xx, No. x, pp. x–x

ALMOST OPTIMAL LOWER BOUNDS FOR PROBLEMS PARAMETERIZED BY CLIQUE-WIDTH*

FEDOR V. FOMIN†, PETR A. GOLOVACH†, DANIEL LOKSHTANOV†, AND SAKET SAURABH‡

Abstract. We obtain asymptotically tight algorithmic bounds for MAX-CUT and EDGE DOMINATING SET problems on graphs of bounded clique-width. We show that on an n-vertex graph of clique-width t both problems

- cannot be solved in time $f(t)n^{o(t)}$ for any function f of t unless Exponential Time Hypothesis (ETH) fails, and
- can be solved in time $n^{O(t)}$.

Key words. Exponential Time Hypothesis, clique-width, max-cut, edge dominating set

AMS subject classifications. 05C85, 68R10, 68Q17, 68Q25, 68W40

1. Introduction.

Tree-width is one of the most fundamental parameters in graph algorithms. Graphs of bounded tree-width enjoy good algorithmic properties similar to trees and this is why many problems which are hard on general graphs can be solved efficiently when the input is restricted to graphs of bounded tree-width. On the other hand, many hard problems also become tractable when restricted to graphs "similar to complete graphs". Courcelle and Olariu [6] introduced the notion of clique-width which captures nice algorithmic properties of both extremes.

Since 2000, the research on algorithmic and structural aspects of clique-width is an active direction in Graph Algorithms, Logic, and Complexity. Corneil et al. [4] show that graphs of clique-width at most 3 can be recognized in polynomial time. Fellows et al. [12] settled a long standing open problem by showing that computing clique-width is NP-hard. Oum and Seymour [31] describe an algorithm that for every fixed t, in time $O(|V(G)|^9 \log |V(G)|)$ computes a $(2^{3t+2}-1)$-expressions of a graph G of clique-width at most t. Recently, Hliněný and Oum obtained time $O(f(t)\cdot|V(G)|^3)$ algorithm computing $(2^{t+1}-1)$-expressions of a graph G of clique-width at most t [19]. We refer to the recent survey [20] for further information on different width parameters.

By the meta-theorem of Courcelle, Makowsky, and Rotics [5], all problems expressible in MS_1-logic (Monadic Second-Order logic on graphs with quantification over subsets of vertices but not of edges) are fixed parameter tractable when parameterized by the clique-width of a graph and the expression size. For many other problems not expressible in this logic including problems like MAX-CUT, EDGE DOMINATING SET, GRAPH COLORING, or HAMILTONIAN CYCLE, there is a significant amount of the literature devoted to algorithms for these problems and their generalizations on graphs of bounded clique-width [10, 16, 17, 18, 25, 26, 28, 33, 34, 35]. Running times of all these algorithms on an n-vertex graph of clique-width at most t are of order $O(n^{f(t)})$, for some functions f of t.

*Extended abstract of results in this paper appeared in the proceedings of SODA 2010 [14].

†Department of Informatics, University of Bergen, N-5020 Bergen, Norway. E-mails: {fedor.fomin,petr.golovach,daniello}@ii.uib.no. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreement n. 267959.

‡The Institute of Mathematical Sciences, CIT Campus, Chennai 600 113, India. E-mail: (saket@imsc.res.in).

One of the central questions in the area is whether the bound $O(n^{f(t)})$ on the running time of all these algorithms is asymptotically optimal. Even the existence of fixed parameter tractable algorithms (with clique-width being the parameter) for all these problems (or their generalizations) was open until very recently [16, 25, 26, 28, 18]. As the first step toward obtaining lower bounds for clique-width parameterizations, we have shown in [15] that unless FPT= W[1], there is no function g such that Graph Coloring, Edge Dominating Set, and Hamiltonian Path are solvable in time $g(t) \cdot n^{O(1)}$. While [15] resolves the parameterized complexity of these problems, the conclusion that unless FPT= W[1], there is no algorithm with run time $O(g(t) \cdot n^c)$, for some function g and a constant c not depending on t, is weak compared to the known algorithmic upper bounds. For example, it does not rule out an algorithm of running time $n^{O(\sqrt{t})}2^t$.

In this paper, we provide asymptotically tight optimal lower bounds for Max-Cut and Edge Dominating Set. In particular, we show that unless Exponential Time Hypothesis (ETH) fails, there is no $f(t)\cdot n^{o(t)}$-time algorithm for both problems. While known algorithms for these problems run in times $n^{O(t^2)}$ [25, 26, 10, 35], we give new algorithmic upper bounds of the form $n^{O(t)}$. Together, these lower and upper bounds give asymptotically tight algorithmic bounds for Max-Cut and Edge Dominating Set.

To obtain our lower bounds we construct "linear FPT-reductions". These type of reductions are much more stringent and delicate than the usual FPT reductions. This is the reason why this research direction is still in a nascent stage and not so many asymptotically tight bounds are known in the literature. Chen, Huang, Kanj, and Xia [2, 3] were the first to succeed in obtaining results of such flavor by showing that there is no algorithm for k-Clique (finding a clique of size k) running in time $f(k)n^{o(k)}$ unless there exists an algorithm for solving 3-SAT running in time $2^{o(n)}$ on a formula with n-variables. The assumption that there does not exists an algorithm for solving 3-SAT running in time $2^{o(n)}$ is known as Exponential Time Hypothesis (ETH) [21] and is equivalent to the following conjecture from parameterized complexity: FPT$\neq$M[1], see [8, 13]. The lower bound on k-Clique can be extended to some other parameterized problems via linear FPT-reductions [2, 3]. This kind of investigation has also been useful in obtaining tight algorithmic lower bounds for polynomial time approximation schemes [29] and for constraint satisfaction problems when parameterized by the tree-width of the "primal graph" [30]. We further extend the utility of this approach by obtaining asymptotically tight algorithmic bounds for clique-width parameterizations. We refer to the recent survey of Lokshtanov, Marx, and Saurabh [27] for detailed discussions on the ETH.

The remaining part of the paper is organized as follows. Section 2 contains definitions and preliminary results. Section 3, is devoted to the proof of an auxiliary result, namely that Red-Blue Capacitated Dominating Set cannot be solved in time $f(k)n^{o(k)}$ on graphs with a feedback vertex set of size at most k assuming ETH. Red-Blue Capacitated Dominating Set and its variants appear to be very handy tools for proving intractability of problems parameterized by the clique-width. In Section 4 we provide tight upper and lower bounds for Max-Cut. We also show how the obtained bound implies bounds on related Bipartization by Edge Removal and Maximum (Minimum) Bisection problems. In Section 5, we obtain bounds for Edge Dominating Set. We conclude by Section 6, where we provide open problems and directions for further research.

2. Definitions and preliminary results.

Parameterized Complexity and ETH.. Parameterized complexity is a two-dimensional framework for studying the computational complexity of a problem. One dimension is the input size n and the other is the *parameter* k. More formally, a (parameterized) language $\mathcal{L} \subseteq \Sigma^* \times \mathbb{N}$ for a finite alphabet Σ is contained in FPT if there is a computable function $f\colon \mathbb{N} \to \mathbb{N}$ such that any instance (Q, k) can be decided with respect to $\mathcal{L}$ in time $f(k) \cdot p(|Q|)$ for some polynomial function $p\colon \mathbb{N} \to \mathbb{N}$. We call such a problem *fixed-parameter tractable*. We refer to the books of Downey and Fellows [9], Flum and Grohe [13] and Niedermeier [32] for a detailed treatment to parameterized complexity.

We define the notion of parameterized (linear) reduction which is the main tool for establishing of our results.

Definition 1. *Let A and B be parameterized problems. We say that A is (uniformly many:1)* FPT-reducible *to B if there exist functions $f, g : \mathbb{N} \to \mathbb{N}$, a constant $\alpha \in \mathbb{N}$ and an algorithm Φ transforming an instance (x, k) of A into an instance $(x', g(k))$ of B in time $f(k)|x|^\alpha$ such that $(x, k) \in A$ if and only if $(x', g(k)) \in B$. The reduction is called* linear *if $g(k) = O(k)$.*

The following well-known complexity hypothesis was formulated by Impagliazzo, Paturi, and Zane [22].

Hypothesis 1 (ETH). *There exists a positive real number s such that* 3Sat *with n variables and m clauses cannot be solved in time $2^{sn}(n+m)^{O(1)}$.*

Graphs.. 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 $E(G)$. A set $S \subseteq V(G)$ of pairwise adjacent vertices is called a *clique*. For $v \in V(G)$, by $E_G(v)$ we denote the set of edges incident with v. For vertex $v \in V(G)$, we denote by $N_G(v)$ its *(open) neighborhood*, that is, the set of vertices adjacent to v. The *closed neighborhood* of v is $N_G[v] := N_G(v) \cup \{v\}$. The *degree* of a vertex v is $d_G(v) := |N_G(v)|$. For graph G, its incidence graph is the bipartite graph $I(G)$ with the vertex set $V(G) \cup E(G)$ such that $v \in V(G)$ and $e \in E(G)$ are adjacent if and only if v is incident with e in G. For a set of vertices $S \subseteq V(G)$, $G[S]$ is the subgraph of G induced by S and by $G - S$ we denote the graph obtained form G by removal of S.

Clique-width.. Let G be a graph, and t be a positive integer. A *t-graph* is a graph with vertices labeled by integers from $\{1, 2, \ldots, t\}$. We refer to a t-graph consisting of exactly one vertex labeled by some integer from $\{1, 2, \ldots, t\}$ as to an *initial t-graph*. The *clique-width* $\mathbf{cwd}(G)$ is the smallest integer t such that G can be constructed by means of repeated application of the following four operations:

- $i(v)$: *Introduce* operation constructing an initial t-graph with vertex v labeled by i,
- $\oplus$: *Disjoint union*,
- $\rho_{i\to j}$: *Relabel* operation changing all labels i to j, and
- $\eta_{i,j}$: *Join* operation making all vertices labeled by i adjacent to all vertices labeled by j.

An *expression tree* of a graph G is a rooted tree T with nodes of four types i, $\oplus$, η and ρ:

- *Introduce nodes* $i(v)$ are leaves of T corresponding to initial t-graphs with vertices v labeled by i.
- *Union node* $\oplus$ stands for a disjoint union of graphs associated with its children.
- *Relabel node* $\rho_{i\to j}$ has one child and is associated with the t-graph obtained by

applying of the relabeling operation to the graph corresponding to its child.
- *Join node* $\eta_{i,j}$ has one child and is associated with the t-graph resulting by applying the join operation to the graph corresponding to its child.
- The graph G is isomorphic to the graph associated with the root of T (with all labels removed).

The *width* of the tree T is the number of different labels appearing in T. If G is of clique-width t, then there is a rooted expression tree T of width t of G. Given a node X of an expression tree, the graph G_X represents the graph formed by the subtree of the expression tree rooted at X.

An expression tree T is *irredundant* if for any join node $\eta_{i,j}$, the vertices labeled by i and j are not adjacent in the graph associated with its child. It was shown by Courcelle and Olariu [6] that every expression tree T of G can be transformed into an irredundant expression tree T' of the same width in time linear in the size of T.

Feedback vertex set.. A *feedback vertex set* of a graph G is a set of vertices $X \subseteq V(G)$ such that $G - X$ is a forest. The *feedback vertex set number* of a graph G, denoted by $\mathbf{fvs}(G)$, is the size of a smallest feedback vertex set of G. We need the following observation.

OBSERVATION 1. *Let X be a feedback vertex set of a graph G. If G' is obtained from G by subdividing some edges, then X is a feedback vertex set of G'.*

We also use the following lemma.

LEMMA 2.1. *Let X be a feedback vertex set of a graph G such that each vertex v of the forest $F = G - X$ is adjacent to at most one vertex of X. Then $\mathbf{cwd}(G) \leq 4 \cdot |X| + 3$.*

Proof. Let $X = \{x_1, \ldots, x_k\}$. Observe that because F is a forest, we have that $\mathbf{cwd}(F) \leq 3$ (see e.g. [24]). Let T be an expression tree for F of width 3 using labels $1, 2, 3$. To construct an expression tree for G, we use the following additional labels.

- Labels $\alpha_1^{(i)}, \ldots, \alpha_k^{(i)}$ for $i \in \{1, 2, 3\}$ for vertices of F adjacent to vertices of X.
- Labels $\beta_1, \ldots, \beta_k$ for vertices of X.

We construct an expression tree for G by making necessary changes in T. For each vertex $v \in V(F)$ adjacent to x_p, the corresponding introduce node $i(v)$ is replaced by node $\alpha_p^{(i)}(v)$. Each relabel node $\rho_{i\to j}$ is replaced by path $\rho_{i\to j}\rho_{\alpha_1^{(i)}\to\alpha_1^{(j)}}\cdots\rho_{\alpha_k^{(i)}\to\alpha_k^{(j)}}$, children of $\rho_{i\to j}$ become children of the first endpoint $\rho_{i\to j}$ of the path, and the parent of $\rho_{i\to j}$ becomes the parent of the second endpoint $\rho_{\alpha_k^{(i)}\to\alpha_k^{(j)}}$. In similar way each join node $\eta_{i,j}$ is replaced by path $\eta_{i,j}\eta_{i,\alpha_q^{(j)}}\eta_{\alpha_p^{(i)},j}\eta_{\alpha_p^{(i)},\alpha_q^{(j)}}$, where indices p, q run through all pairs of different integers from $\{1, \ldots, k\}$. Then we construct an expression tree for $G[X]$ with introduce nodes $\beta_1(x_1), \ldots, \beta_k(x_k)$. Since all vertices of X are labeled by pairwise different labels, this construction is done in a straightforward way. Finally, we add the union vertex with the roots of the modified tree for F and the tree for $G[X]$ being its children, and construct the join nodes $\eta_{\alpha_1^{(i)},\beta_1}, \ldots, \eta_{\alpha_k^{(i)},\beta_k}$ for $i \in \{1, 2, 3\}$. □

3. Capacitated domination. In this section we prove an auxiliary theorem about the CAPACITATED DOMINATING SET problem. This result will be heavily used in the proof of the main results of this paper. The parameterized complexity of CAPACITATED DOMINATING SET with the treewidth of the input graph being the parameter was considered in [1, 7]. Here, we use a special variant of the problem and parameterize it by the feedback vertex set number.

A *red-blue capacitated graph* is a pair (G, c), where G is a bipartite graph with the vertex bipartition R and B and $c\colon R \to \mathbb{N}$ is a *capacity* function such that $1 \leq$

$c(v) \leq d_G(v)$ for every vertex $v \in R$. The vertices of R are called *red* and the vertices of B are called *blue*. A set $S \subseteq R$ is called a *capacitated dominating set* if there is a *domination mapping* $f\colon B \to S$ mapping every vertex from B to one of its neighbors in S such that the total number of vertices mapped by f to each vertex $v \in S$ does not exceed its capacity $c(v)$. For $v \in S$, we say that vertices in $f^{-1}(v)$ are *dominated by* v. The Red-Blue Capacitated Dominating Set (or Red-Blue CDS) problem for a given red-blue capacitated graph (G, c) and a positive integer k, asks whether there exists a capacitated dominating set S for G containing at most k vertices. A capacitated dominating set $S \subseteq R$ is called *saturated* if there is a domination mapping f which saturates all vertices of S, that is, $|f^{-1}(v)| = c(v)$ for each $v \in S$. The Red-Blue Exact Saturated Dominating Set problem (Red-Blue Exact Saturated CDS) takes a red-blue capacitated graph (G, c) and a positive integer k as an input and asks whether there exists a saturated capacitated dominating set with exactly k vertices.

The main result of this section is the following technical theorem.

Theorem 3.1. *Unless the ETH fails,* Red-Blue CDS *and* Red-Blue Exact Saturated CDS *cannot be solved in time $f(k)n^{o(k)}$, where n is the number of vertices and k is the feedback vertex set number of the input graph. Moreover, none of the two problems can be solved in time $f(k)n^{o(k)}$ even if the input is restricted to graphs G such that for every minimum feedback vertex set $X \subseteq V(G)$,*

- *X is independent, and*
- *each vertex of the forest $G - X$ is adjacent to at most one vertex of X.*

The remaining part of the section is devoted to the proof of this theorem.

Proof. [Proof of Theorem 3.1] First we prove the claim for Red-Blue CDS. We construct a linear FPT-reduction from the k-Multi-Colored Clique (k-MCC) problem to Red-Blue CDS parameterized by the feedback vertex set number. In fact, our reduction runs in polynomial time.

The k-MCC problem asks for a given k-partite graph $G = (V_1 \cup \cdots \cup V_k, E)$, where $V_1, \ldots, V_k$ are sets of the k-partition, whether there is a k-clique C in G. The fact that assuming ETH this problem can not be solved in time $f(k)n^{o(k)}$, follows immediately from the reduction from the k-Clique problem (see e.g. [11, 27]). We show how for a given instance (G, k) of k-MCC to construct an instance (H, c, k') of Red-Blue CDS such that

- $\mathbf{fvs}(H) = 2k$,
- $k' = \binom{k}{2} + 5k$,
- G has a clique of size k if and only if H has a capacitated dominating set of size k', and
- for every minimum feedback vertex set $X \subseteq V(H)$, X is independent and each vertex of the forest $H - X$ is adjacent to at most one vertex of X.

Let us note that while the number of vertices in a capacitated dominating set of H is $\mathcal{O}(k^2)$, we have $\mathbf{fvs}(H) = 2k$ and thus this reduction is linear for the required parameterization of Red-Blue CDS. Without loss of generality we assume that $k \geq 3$.

To each vertex $v \in V(G)$ we will assign two unique identification numbers v^{up} and v^{down}. The choice of identification numbers plays a crucial role in our reduction. Let n be the number of vertices in G. For an integer p, we say that a set X of positive integers is a *p-non-averaging* set if for every tuple $(x_1, x_2, \ldots, x_p)$ of elements of X their average is in X if and only if $x_1 = x_2 = \cdots = x_p$. Remarkably, dense p-

non-averaging sets exist, and can be constructed in polynomial time.

LEMMA 3.2 ([23]). *For every p and n there exists a p-non-averaging set X, $|X| = n$, such that the largest element of X has value $32p^2n^2$. Furthermore, X can be constructed in $O(p^2n^3)$ time.*

We construct a $(k-1)$-non-averaging set X of size n and assign to each vertex v of G a unique identification number $v^{up} \in X$. Now we fix t to be three times the largest element of X. For every v, we set $v^{down} = t - v^{up}$.

For two red vertices u and v of H and a positive integer A, by adding an *A-arrow* from u to v we will mean adding A subdivided edges between u and v. All the vertices on the added subdivided edges are blue. Now we describe how to build the graph H for a given instance $(G = (V_1 \cup V_2 \cdots \cup V_k, E), k)$ of k-MCC.

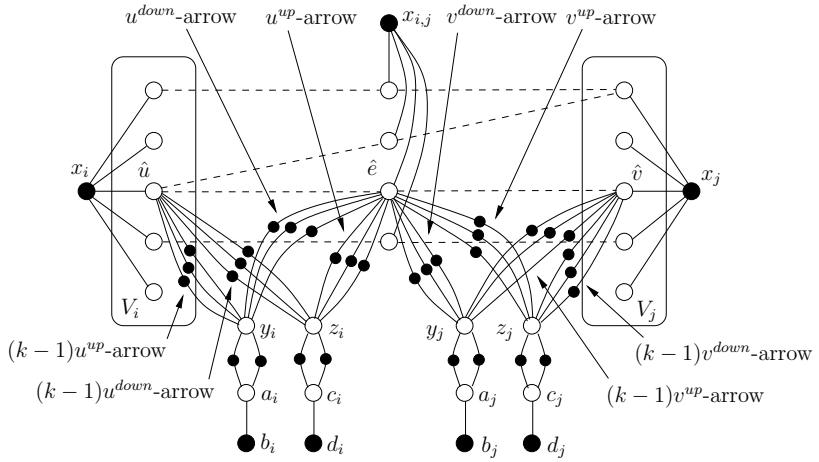

FIG. 1. *Partial construction of H corresponding to two sets V_i and V_j of the k-MCC instance. The edges of the original graph are shown by dashed lines; the vertices from R are shown in white and vertices of B are shown in black.*

For a pair of distinct integers $i, j \in \{1, \ldots, k\}$, let $E_{i,j}$ be the set of edges with one endpoint in V_i and the other in V_j. For every integer i between 1 and k, we make a blue vertex x_i that has a red neighbor $\hat{v}$ for each $v \in V_i$. For every pair of integers i,j such that $1 \leq i < j \leq k$, we make a blue vertex $x_{i,j}$ that has a red neighbor $\hat{e}$ for every edge e in $E_{i,j}$. For every i, we add a red vertex y_i and a red vertex z_i, which we call *marked*. For every marked vertex y_i, a red vertex a_i and a blue vertex b_i adjacent to a_i are added, and for y_i and a_i, a 2-arrow is constructed. Similarly, for every marked vertex z_i, a red vertex c_i and a blue vertex d_i adjacent with c_i are added, and z_i is joined with c_i by a 2-arrow.

Now, for every vertex $v \in V_i$ we add a $((k-1) \cdot v^{up})$-arrow from $\hat{v}$ to y_i and a $((k-1) \cdot v^{down})$-arrow from $\hat{v}$ to z_i. For every $1 \leq i < j \leq k$ and edge $e = uv$ in $E_{i,j}$ we proceed as follows. Let $u \in V_i$ and $v \in V_j$. We add a (u^{down})-arrow from $\hat{e}$ to y_i, a (u^{up})-arrow from $\hat{e}$ to z_i, a (v^{down})-arrow from $\hat{e}$ to y_j and a (v^{up})-arrow from $\hat{e}$ to z_j. At this point, if any marked vertex has degree less than $(k-1)t+2$, we add blue leaves adjacent only to that vertex, such that the marked vertex gets degree $(k-1)t+2$. This concludes the construction of H. The construction is shown in Fig. 1. Finally, we describe the capacities of the red vertices. We set capacities of all vertices a_i and c_i equal to one. All other red and unmarked vertices have capacities equal to their degree. The marked vertices all have capacity exactly $(k-1)t$ less than

their degree.

It is easy to see that $2k$ marked vertices form a feedback vertex set of H. Moreover, all marked vertices should be contained in each feedback vertex set of a size at most $2k$ because of 2-arrows which join these vertices with the vertices a_i and c_i respectively. Indeed, let X be a feedback vertex set of size at most $2k$. Because 2-arrows form $2k$ vertex disjoint cycles of length 4, set X should contain at least one vertex from each of these cycles. Hence, $|X| = 2k$ and thus only vertices of these 2-arrows can be in X. Now if there was a marked vertex, say y_i not in X, then because $k \geq 3$, for each $v \in V_i$, the $((k-1) \cdot v^{up})$-arrow from $\hat{v}$ to y_i contains a cycle. On the other hand, X cannot contain vertices of this cycle; a contradiction. Hence, $\mathbf{fvs}(H) = 2k$, and the set of the marked vertices is a unique minimum feedback vertex set.

By the construction, the set of marked vertices is independent and each marked vertex is adjacent only to subdivision vertices of A-arrows. Therefore, each vertex of the forest obtained from H by removal of vertices of the feedback vertex set is adjacent to at most one vertex of this set.

The following two lemmata complete the proof.

LEMMA 3.3. *If G has a multicolor clique $C = \{v_1, v_2, \ldots, v_k\}$, then H has a capacitated dominating set D of size k'.*

Proof. For every $i < j$ let e_{ij} be the edge from v_i to v_j in G. In addition to all the marked vertices and vertices a_i and c_i for $i \in \{1, \ldots, k\}$, let D contain $\hat{v_i}$ and $\hat{e_{ij}}$ for every $i, j \in \{1, \ldots, k\}$, $i < j$. Clearly D contains exactly k' vertices, so it remains to prove that D is indeed a capacitated dominating set.

Vertices a_i and c_i dominate blue neighbors b_i and d_i respectively. All other unmarked red vertices have degree equal to their capacity, so such vertices in D dominate all their neighbors. Thus, all the x_i-s are dominated by unmarked vertices in D. Observe that every blue vertex except for the x_i, b_i and d_i-s has exactly one marked neighbor. Thus, since the marked vertices all have capacity exactly $(k-1)t$ less than their degree, it is sufficient to prove that every marked vertex has at least $(k-1)t$ blue neighbors that are dominated by unmarked vertices in D.

Consider y_i for some i. Notice that $\hat{v_i}$ dominates $(k-1) \cdot v_i^{up}$ blue neighbors of y_i. Also, every $\hat{e_{ij}}$ such that $i < j$ and every $\hat{e_{ji}}$ such that $j < i$ dominates v_i^{down} blue neighbors of y_i. Thus $(k-1)(v_i^{up} + v_i^{down}) = (k-1)t$ blue neighbors of y_i are dominated by unmarked vertices in D. The proof for z_i is identical. Namely, $\hat{v_i}$ dominates $(k-1) \cdot v_i^{down}$ blue neighbors of z_i. Also, every $\hat{e_{ij}}$ such that $i < j$ and every $\hat{e_{ji}}$ such that $j < i$ dominates v_i^{up} blue neighbors of z_i. Thus $(k-1)(v_i^{down} + v_i^{up}) = (k-1)t$ blue neighbors of z_i are dominated by unmarked vertices in D. This concludes the proof. □

LEMMA 3.4. *If H has a capacitated dominating set D of size k', then G has a multicolor clique of size k.*

Proof. For each $i \in \{1, \ldots, k\}$, a_i has capacity one and has a private neighbor b_i which should be dominated. Therefore y_i must be included in D and must dominate two adjacent vertices in the 2-arrows joining y_i and a_i. Similarly, every z_i must be included in D and must dominate two adjacent vertices in the 2-arrows which joins z_i and c_i.

For every $i \in \{1, \ldots, k\}$, there is $v_i \in V_i$ such that $\hat{v}_i \in D$, because otherwise x_i is undominated. Similarly, for every pair of integers i, j with $i < j$ there must be an edge $e_{ij} \in E_{i,j}$ such that $\hat{e}_{ij} \in D$; otherwise x_{ij} is undominated. Also all vertices b_i and d_i should be dominated, and hence all vertices a_i and c_i must be included in D. Since $|D| \leq k'$ it follows that these are the only unmarked vertices in D.

Since all unmarked vertices in D except a_i, and c_i, $i \in \{1,\ldots,k\}$, have capacity equal to their degree, we can assume that each such vertex dominates all its neighbors. For $j > i$, define $e_{ji} = e_{ij}$ and $\hat{e}_{ji} = \hat{e}_{ij}$. We proceed by proving that for every pair of integers i,j with $i \neq j$, $e_{ij} = uv$ is incident with v_i.

Consider z_i for some i. The blue neighbors of z_i can be dominated by z_i, $\hat{v}_i$ or some $\hat{e}_{ij}$ for $j \neq i$. Since the capacity of z_i is $(k-1)t$ less than its degree we have that at least $(k-1)t$ neighbors of z_i are dominated by some vertices $\hat{v}_i$ or $\hat{e}_{ij}$ in D. For every $j \neq i$, let u_i be the endpoint of e_{ij} that is in V_i. Now $\hat{v}_i$ dominates $(k-1)v_i^{down}$ neighbors of z_i and for every $j \neq i$ we have that $\hat{e}_{ij}$ dominates u_j^{up} neighbors of z_i. Hence,

$$(k-1)v_i^{down} + \sum_{j \neq i} u_j^{up} \geq (k-1)t. \tag{3.1}$$

An identical argument for y_i yields $(k-1)v_i^{up} + \sum_{j\neq i} u_j^{down} \geq (k-1)t$. For every $v \in V(G)$ we have $v^{up} + v^{down} = t$ and thus

$$(k-1)v_i^{down} + \sum_{j \neq i} u_j^{up} + (k-1)v_i^{up} + \sum_{j \neq i} u_j^{down} = 2(k-1)t.$$

Thus it follows that the inequality in (3.1) is the equality, yielding the following

$$\sum_{j \neq i} u_j^{up} = (k-1)t - (k-1)v_i^{down} = (k-1)v_i^{up}. \tag{3.2}$$

Since the up-identification numbers were taken from a $(k-1)$-non-averaging set it follows that $u_j = v_i$ for all $j \neq i$. Thus for every i and $j \neq i$, v_i is incident with e_{ij}. Thus $v_1,\ldots,v_k$ form a multi-colored clique in G. This concludes the proof. □

To prove the claim of Theorem 3.1 for Red-Blue Exact Saturated CDS, it is sufficient to observe that if the instances constructed in the proof for Red-Blue CDS have a capacitated dominating set D of size k', then the capacitated dominating set D is saturated and exact. □

4. Max-Cut and related problems. In this section we consider the Max-Cut problem and a few other closely related problems. Let G be a graph. For a partition V_1, V_2 of $V(G)$, the *cut set* is defined as $C_G(V_1,V_2) = \{uv \in E(G) \colon u \in V_1, v \in V_2\}$. A *cut set* of a graph G is a set of edges $C \subseteq E(G)$ such that there is a partition V_1, V_2 of $V(G)$ with $C = C_G(V_1,V_2)$. The size of a maximum cut set in G is denoted by $\mathbf{mcut}(G)$. In the Max-Cut problem, we are given a graph G and a positive integer k, and the objective is to check whether there exists a cut set $C \subseteq E(G)$ such that $|C| \geq k$. Our main theorem in this section is the following.

Theorem 4.1. *Let G be an n-vertex graph given together with an expression tree of width t. Then the* Max-Cut *problem*

- *cannot be solved in time $f(t) \cdot n^{o(t)}$ unless the ETH fails;*
- *is solvable in time $n^{O(t)}$.*

We prove this theorem in two steps. We first show the lower bound and then complement this result with the corresponding upper bound.

4.1. Lower bound. To prove the lower bound we give a reduction from the Red-Blue CDS problem parameterized by the feedback vertex set number to the Max-Cut problem. The proof is organized as follows: we first give a construction, then prove its correctness and finally argue on the clique-width of the transformed instance.

Construction.. Let (G, c, k) be an instance of Red-Blue CDS with $R = \{u_1, \ldots, u_n\}$ being the set of red vertices and $B = \{v_1, \ldots, v_r\}$ being the set of blue vertices, and such that for every minimum feedback vertex set X of G, set X is independent. We also assume that G has m edges.

We start from the auxiliary gadgets.

Auxiliary gadgets $F(x, y)$ ***and*** $F'(x, y)$*..* Let x, y be two vertices. We construct $F(x, y)$ by joining x and y by $4m+1$ paths of length two. Graph $F'(x, y)$ is constructed by joining x and y by $4m + 1$ paths of length three. The properties of $F(x, y)$ and $F'(x, y)$ required for our proof are summarized in the following lemma.

Lemma 4.2. *For every $F(x, y)$ and $F'(x, y)$*

- $\mathbf{mcut}(F(x, y)) = 8m + 2$ *and* $\mathbf{mcut}(F'(x, y)) = 12m + 3$.
- *For every partition V_1, V_2 of the vertex set of $F(x, y)$ such that $x \in V_1$ and $y \in V_2$, $|C_{F(x,y)}(V_1, V_2)| \leq \mathbf{mcut}(F(x, y)) - 4m - 1$.*
- *For every partition V_1, V_2 of the vertex set of $F'(x, y)$ such that $x, y \in V_1$, $|C_{F'(x,y)}(V_1, V_2)| \leq \mathbf{mcut}(F'(x, y)) - 4m - 1$.*

We will attach gadgets $F(x, y)$ and $F'(x, y)$ to other parts of the construction through the vertices x and y. Notice that we can always assume that the vertices of $V(F(x, y))\setminus\{x, y\}$ are included in exactly one side of an optimal partition of the vertex set leading to the maximum sized cut. Similarly, we can assume that the vertices of $N_{F'(x,y)}(x)$ ($N_{F'(x,y)}(y)$ respectively) also included in exactly one side of an optimal partition of the vertex set.

Auxiliary gadgets $H_{s,t}(x_1, \ldots, x_s, y)$*..* Let $\ell = \max\{n, r\}$. We first construct graph H with the vertex set $\{z_{i,j} \colon 1 \leq i \leq 2\ell, 1 \leq j \leq 4m + 1\}$. Vertices $z_{i,j}$ and $z_{i',j'}$ are joined by an edge for $1 \leq i < i' \leq 2\ell$. That is, we construct a complete 2ℓ-partite graph with the 2ℓ-partition $Z_1, \ldots, Z_{2\ell}$, where $Z_i = \{z_{i,1}, \ldots, z_{i,4m+1}\}$. Then we add graphs $F(z_{i,1}, z_{i,2}), \ldots, F(z_{i,4m}, z_{i,4m+1})$ for each $i \in \{1, \ldots, 2\ell\}$. Let $h = \ell(4m + 1)(4m\ell + 16m + \ell)$. We observe that a partition V_1, V_2 corresponding to $\mathbf{mcut}(H)$ is the following. Let V_1 consists of $Z_1, \ldots, Z_\ell$ and all the vertices of gadgets $F(z_{i,1}, z_{i,2}), \ldots, F(z_{i,4m}, z_{i,4m+1})$, $i \in \{\ell + 1, \ldots, 2\ell\}$, except those vertices of gadgets contained in $Z_{\ell+1}, \ldots, Z_{2\ell}$. Let V_2 be the remaining vertices. Using partition V_1, V_2 corresponding to $\mathbf{mcut}(H)$, by Lemma 4.2, we obtain the following lemma.

Lemma 4.3. *For every partition V_1, V_2 of the vertex set of H, we have the following. If V_1 (or V_2) does not contain exactly ℓ sets from $Z_1, \ldots, Z_{2\ell}$, then $|C_H(V_1, V_2)| \leq \mathbf{mcut}(H) - 4m - 1$. Furthermore, $\mathbf{mcut}(H) = h$.*

Let s and t be two positive integers such that $s, t \leq \ell$. We construct graph $H_{s,t}(x_1, \ldots, x_s, y)$ from H by adding vertices $x_1, \ldots, x_s$ and y, and then joining them with H by gadgets $F(x_1, z_{1,1}), \ldots, F(x_s, z_{s,1})$ and $F(y, z_{\ell+1,1}), \ldots, F(y, z_{\ell+t,1})$ (see Fig 2). Let $h_{s,t} = h + (8m + 2)(s + t)$. We say that the subgraph of $H_{s,t}(x_1, \ldots, x_s, y)$ induced by Z_i and the set vertices of degree two adjacent to the vertices of Z_i is the *i-th column of $H_{s,t}(x_1, \ldots, x_s, y)$* for $i \in \{1, \ldots, 2\ell\}$. For $i \in \{1, \ldots, s\}$, we say that the i-th column is *associated with x_i*. Respectively, for $i \in \{\ell + 1, \ldots, \ell + t\}$, the i-th column is *associated with y*. We also refer to vertices $z_{i,j}$ as to *z-vertices* of the gadget. Lemmata 4.2 and 4.3 imply the following properties of this graph.

Lemma 4.4. *The following properties hold for $H_{s,t}(x_1, \ldots, x_s, y)$.*

- $\mathbf{mcut}(H_{s,t}(x_1, \ldots, x_s, y)) = h_{s,t}$.
- *Let V_1, V_2 be an optimal partition of $V(H_{s,t}(x_1, \ldots, x_s, y))$, that is, $\mathbf{mcut}(H_{s,t}(x_1, \ldots, x_s, y)) = |C_{H_{s,t}(x_1,\ldots,x_s,y)}(V_1, V_2)|$, and $y \in V_1$. Then at most $\ell - t$ vertices from $\{x_1, \ldots, x_s\}$ are in V_1.*
- *There is an optimal partition V_1, V_2 such that $y \in V_1$ and for each $0 \leq p \leq$*

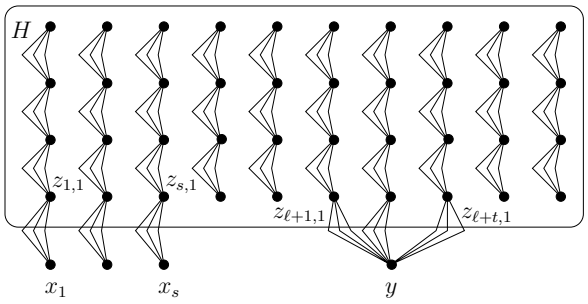

FIG. 2. *Graph* $H_{s,t}(x_1,\ldots,x_s,y)$

$\min\{s,\ell-t\}$, *exactly* p *vertices from* $\{x_1,\ldots,x_s\}$ *are in* V_1.

- *For every non-optimal partition* V_1,V_2 *of* $V(H_{s,t}(x_1,\ldots,x_s,y))$, *with the following two properties*
  - (a) *for every gadget* $F(z_{i,j},z_{i,j+1})$, $1\leq i\leq 2\ell$ *and* $1\leq j\leq 4m$, *we have that* $V(F(z_{i,j},z_{i,j+1}))\setminus\{z_{i,j},z_{i,j+1}\}$ *is contained either in* V_1, *or* V_2;
  - (b) *for every gadget* $F(x_i,z_{i,1})$, $1\leq i\leq s$ *and* $F(y,z_{\ell+j,1})$ $1\leq j\leq t$, *we have that* $V(F(x_i,z_{i,1}))\setminus\{x_i,z_{i,1}\}$ *and* $V(F(y,z_{\ell+j,1}))\setminus\{y,z_{\ell+j,1}\}$ *is contained either in* V_1, *or* V_2,

  we have that $|C_{H_{s,t}(x_1,\ldots,x_s,y)}(V_1,V_2)|\leq \mathbf{mcut}(H_{s,t}(x_1,\ldots,x_s,y))-4m-1$.

Final Reduction.. We are ready to describe the reduction. Each edge $e=u_iv_j$ of G is replaced by two vertices a_e and b_e. Each of the new vertices becomes adjacent to u_i and v_j. Thus we replace edge e with two paths of length two. We create two vertices w_1 and w_2 and construct a copy of $F'(w_1,w_2)$. For each vertex $v_j\in B$, a copy of $F(v_j,w_1)$ is created. In the next step, we introduce a copy of $H_{n,\ell-k}(u_1,\ldots,u_n,w_1)$. By G' we denote the graph obtained until now (we will need this graph while bounding the clique-width of the construction in Lemma 4.6). Finally, for each vertex $u_i\in R$, a copy of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$, where the set of edges incident with u_i, $E_G(u_i)=\{e_1,\ldots,e_{d_G(u_i)}\}$ is constructed, and for each vertex $v_j\in B$, a copy of $H_{d_G(v_j),\ell-1}(a_{e_1},\ldots,a_{e_{d_G(v_j)}},w_2)$, where $\{e_1,\ldots,e_{d_G(v_j)}\}=E_G(v_j)$ is added. Let Q be the resulting graph. We put

$$\mu=(4m+1)(2r+3)+h_{n,\ell-k}+\sum_{i=1}^{n}h_{d_G(u_i),\ell-c(u_i)}+\sum_{j=1}^{r}h_{d_G(v_j),\ell-1}+2(m+r).$$

LEMMA 4.5. *Graph* G *has a capacitated dominating set of the size at most* k *if and only if graph* Q *has a cut set with at least* μ *edges.*

Proof. Let S be a capacitated dominating set of size at most k in G and f be the corresponding domination mapping. We construct a partition V_1,V_2 of the vertex set of Q corresponding to a cut set of size at least μ as follows. We put vertex w_1 in V_1, vertex w_2 in V_2, all vertices $v_1,\ldots,v_r$ in V_1, all vertices from S in V_1, and and vertices of $R\setminus S$ in V_2. We also put all vertices b_e in V_2. For each edge $e=u_iv_j\in E(G)$ such that $f(v_j)=u_i$, that is, e is being used for domination, the corresponding vertex a_e is included in V_2 and all other vertices a_e, whose corresponding edge is not used for domination, are included in V_1. Finally, we extend our partition to an optimal partition of all gadgets $F(x,y)$, $F'(x,y)$ and $H_{s,t}(x_1,\ldots,x_s,y)$ used in the construction of Q. Desired extensions of these gadgets to an optimal partition can be done by applications of Lemmata 4.2, 4.3 and 4.4. By the construction of partitions

V_1 and V_2, the contribution of gadgets $F(x,y)$, $F'(x,y)$ and $H_{s,t}(x_1,\ldots,x_s,y)$ to the cut $C_Q(V_1,V_2)$ is $\mathbf{mcut}(F(x,y))$, $\mathbf{mcut}(F'(x,y))$, and $\mathbf{mcut}(H_{s,t}(x_1,\ldots,x_s,y))$ respectively. Hence, we have already accounted for

$$(4m+1)(2r+3)+h_{n,\ell-k}+\sum_{i=1}^{n}h_{d_G(u_i),\ell-c(u_i)}+\sum_{j=1}^{r}h_{d_G(v_j),\ell-1}$$

edges in the cut $C_Q(V_1,V_2)$. The remaining $2(m+r)$ edges in the cut $C_Q(V_1,V_2)$ come from the edges incident with vertices a_e and b_e for some e. Every edge $e=uv$ is either it is an edge used for domination or not. In the first case, when e is used for dominating, we have that ua_e, a_ev, ub_e, and b_ev are part of the cut. In the second case, exactly two of the edges among ua_e, a_ev, ub_e, and b_ev are part of the cut. In each case for every e at least two edges among ua_e, a_ev, ub_e and b_ev are in the cut and hence edges incident with vertices a_e and b_e contribute at least $2(m-r)+4r=2(m+r)$ to the cut $C_Q(V_1,V_2)$. This completes the forward direction of the proof.

Assume now that Q has a cut set C of size at least μ, and let (V_1,V_2) be the corresponding partition of the vertex set of Q. Let Q' be the graph obtained by taking the union of the edge sets of auxiliary gadgets $F(x,y)$, $F'(x,y)$ and $H_{s,t}(x_1,\ldots,x_s,y)$. Then there exists a partition A and B of $V(Q')$ such that $C_{Q'}(A,B)=\mu'$, where

$$\mu'=(4m+1)(2r+3)+h_{n,\ell-k}+\sum_{i=1}^{n}h_{d_G(u_i),\ell-c(u_i)}+\sum_{j=1}^{r}h_{d_G(v_j),\ell-1}.$$

Suppose that at least for one of the gadgets $F(x,y)$, $F'(x,y)$, or $H_{s,t}(x_1,\ldots,x_s,y)$, say $F(x,y)$, the partition (V_1',V_2') of $V(F(x,y))$ obtained by restricting the partition (V_1,V_2) to $V(F(x,y))$ is not optimal. That is, $|C_{F(x,y)}(V_1',V_2')|<\mathbf{mcut}(F(x,y))$. Then by Lemmata 4.2, 4.3, and 4.4, $|C|\leq\mu'-(4m+1)+4m<\mu$. By choosing non-optimal partitions of auxiliary gadgets we loose at least $4m+1$ edges while gaining at most $4m$ new edges by cutting $4m$ edges of Q which do not belong to these gadgets. This implies that C restricted to all these gadgets is an optimal cut in Q'. By Lemma 4.2, w_1 and w_2 belong to different sets of the bipartition V_1,V_2. Assume that $w_1\in V_1$ and $w_2\in V_2$. Then Lemma 4.2 implies that $v_1,\ldots,v_r\in V_1$. Thus, by making use of Lemma 4.4, we conclude that at most k vertices of set $R=\{u_1,\ldots,u_n\}$ belong to V_1. We put $S=R\cap V_1$ and prove that S is a capacitated dominating set in G. Notice that by Lemma 4.4, at most one vertex a_e in the neighborhood of each vertex v_j is included in V_2. Suppose that there is a vertex v_j such that its neighborhood in Q has no vertices $a_e\in V_2$. Then $|C|\leq\mu'+2m+2(r-1)<\mu$, a contradiction. So, for each vertex v_j, there is an edge $e=u_iv_j$ such that $a_e\in V_2$. Now we argue that $u_i\in S$. This follows from the fact that if $u_i\notin S$, then $u_i\in V_2$; hence $|C|\leq\mu'+2m+2r-2<\mu$. We define the domination mapping as $f(v_j)=u_i$. Since by Lemma 4.4, at most $c(u_i)$ vertices in the set $N_Q(u_i)\cap\{a_e\mid e\in E(G)\}$ are included in V_2, we have that $|f^{-1}(u_i)|\leq c(u_i)$. This concludes the proof. □

Now we upper bound the clique-width of Q by a linear function of the feedback vertex set number of G.

LEMMA 4.6. $\mathbf{cwd}(Q)\leq 40\cdot\mathbf{fvs}(G)+40$.

Proof. Let $t=\mathbf{fvs}(G)$ and let X be a minimum feedback vertex set of G. By Observation 1, X is a feedback vertex set of $I(G)$. Recall that X is independent. Then each vertex of $I(G)-X$ is adjacent to at most one vertex of X. By Lemma 2.1, $\mathbf{cwd}(I(G))\leq 4\cdot|X|+3=4t+3$.

Let us remind, that while describing graph Q, as an intermediate step of the construction, we defined graph G'. We construct an expression tree for Q in two steps and use $10c+10$ labels, where $c = 4t+3$. At the first step, we construct en expression tree for G' using $4c+10$ labels, and at the second step we describe how it can modified to an expression tree for Q by the cost of $6c$ additional labels.

Expression tree for G'.. Suppose that an expression tree for $I(G)$ uses c labels $\alpha_1, \ldots, \alpha_c$. To construct an expression tree for G', besides labels $\alpha_1, \ldots, \alpha_c$ we introduce the following additional labels.

- Labels $\beta_1, \ldots, \beta_c$ for vertices $v_1, \ldots, v_r$.
- Labels $\gamma_1, \ldots, \gamma_c$ for vertices $\{a_e \mid e \in E(G)\}$.
- Labels $\delta_1, \ldots, \delta_c$ for vertices $\{b_e \mid e \in E(G)\}$.
- Labels ζ_1, ζ_2 for vertices w_1, w_2.
- Label η for vertices $z_{i,j}$ in $H_{n,\ell-k}(u_1, \ldots, u_n, w_1)$.
- Working labels $\lambda_1, \lambda_2, \lambda_3$ and $\xi_1, \xi_2, \xi_3, \xi_4$.

We construct the required expression tree for G' by going over the expression tree for $I(G)$ and making necessary changes in it. When a vertex $u_i \in R$ labeled by α_p is introduced, we construct u_i and the column of the gadget $H_{n,\ell-k}(u_1, \ldots, u_n, w_1)$ associated with u_i together with the edges that join the column and u_i. To do it, we perform the following set of operations. We first introduce vertex u_i labeled by α_p and a vertex (which is essentially $z_{i,1}$ of the gadget) labeled by ξ_3. Then $4m+1$ vertices labeled by ξ_2 are introduced and joined with vertices labeled by α_p and ξ_3. Then vertices labeled ξ_2 are relabeled to λ_1. Now we repeat the following operations $4m$ times:

- introduce a vertex labeled by ξ_1 and $4m+1$ vertices labeled by ξ_2;
- join vertices labeled by ξ_2 with vertices labeled by ξ_1 and ξ_3;
- relabel vertices labeled by ξ_2 by λ_1, vertex labeled by ξ_3 by η, and vertex labeled by ξ_1 by ξ_3;
- finally, vertex labeled by ξ_3 is relabeled by η.

We omit the union operations from our descriptions here and henceforth in any similar descriptions and assume that if some vertex is introduced then the union is always performed.

When a vertex $x \in V(I(G))$ corresponding to an edge $e \in E(G)$ and labeled by α_p is introduced, we introduce vertices a_e and b_e and label them by γ_p and δ_p, respectively. Now we move towards the introduction of vertices from set B. When a vertex $v_j \in B$ labeled by α_p is introduced, we introduce vertex v_j with label β_p. Then $4m+1$ vertices labeled by ξ_1 are introduced, joined with the vertex labeled by β_p. Then we label these vertices by λ_2 and finally join them with vertex w_1, after w_1 is introduced.

For each union operation in the expression tree for $I(G)$, we do as follows. If both graphs contain vertices labeled η, then both graphs contain columns of $H_{n,\ell-k}(u_1, \ldots, u_n, w_1)$, and the z-vertices from different parts should be joined by edges. To implement this, besides the union operation, we do the following:

- vertices labeled by η in one of the graphs are relabeled by ξ_1;
- we perform the union operation;
- the vertices labeled by η and ξ_1 are joined; and
- the vertices labeled by ξ_1 are relabeled by η.

If only one graph contains vertices labeled η then we just do the union operation.

If in the expression tree of $I(G)$, we have join operation between two labels, say α_p and α_q, then we simulate this by applying join operations between the vertices

with the following labels: α_p and γ_q; α_p and δ_q; β_p and γ_q; β_p and δ_q; α_q and γ_p; α_q and δ_p; β_q and γ_p; and β_q and δ_p.

Finally, for the relabel operation in the expression tree of G, that is, relabel α_p to α_q, is replaced by the following relabeling process: α_p to α_q; β_p to β_q; γ_p to γ_q; and δ_p to δ_q.

After we have scanned the expression tree for $I(G)$, vertices w_1 and w_2 labeled by ζ_1 and ζ_2 respectively, are introduced. Then we repeat the following operations $4m+1$ times:

- introduce two vertices labeled by ξ_1 and ξ_2;
- join vertices labeled by ζ_1 and ξ_1, ξ_1 and ξ_2, ξ_2 and ζ_2;
- and relabel the vertices labeled by ζ_1 and ζ_2 by λ_1.

After that the vertex w_1 labeled by ζ_1 is joined with the vertices labeled λ_2.

We show next how to complete the construction of $H_{n,\ell-k}(u_1,\ldots,u_n,w_1)$. Notice that the columns associated with $u_1,\ldots,u_n$ and the edges that join $u_1,\ldots,u_n$ with these columns are constructed, all z-vertices in distinct columns are already pairwise adjacent, and all z-vertices are labeled by η. Now we construct columns of gadgets that are not associated with $u_1,\ldots,u_n,w_1$ and joining them together. We repeat the following $\ell-n+k$ times. A vertex labeled ξ_3 is introduced, and now we repeat the following operations $4m$ times:

- introduce a vertex labeled ξ_1 and $4m+1$ vertices labeled ξ_2;
- join vertices labeled ξ_2 and vertices labeled with ξ_1 and ξ_3;
- relabel vertices labeled ξ_2 by λ_1, the vertex labeled ξ_3 by ξ_4, and the vertex labeled ξ_1 by ξ_3.

Finally, the vertex labeled ξ_3 is relabeled ξ_4, the vertices labeled ξ_4 are joined with vertices labeled η, and then relabeled by η.

It remains to consider the columns associated with w_1. We do the following $\ell-k$ times. The vertex labeled by ξ_3 and $4m+1$ vertices labeled by ξ_1 are introduced. Vertices labeled by ξ_1 are joined with vertices labeled by ζ_1 and ξ_2, and relabeled λ_1. After this we repeat the following operations $4m$ times:

- introduce new vertex labeled by ξ_1 and $4m+1$ vertices labeled by ξ_2;
- join vertices labeled by ξ_2 and vertices labeled by ξ_1 and ξ_3;
- relabel vertices labeled by ξ_2 by λ_1, vertex labeled by ξ_3 by ξ_4, and the vertex labeled by ξ_1 by ξ_3.

Finally, the vertex labeled ξ_3 is relabeled by ξ_4, the vertices labeled ξ_4 are joined with vertices labeled η and then relabeled η.

Expression tree for Q.. We now show how to modify the expression tree for G' by adding gadgets $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$, where $\{e_1,\ldots,e_{d_G(u_i)}\} = E_G(u_i)$, for $u_i \in R$ by making use of $3c$ additional labels. Gadgets $H_{d_G(v_j),\ell-1}(a_{e_1},\ldots,a_{e_{d_G(v_j)}},w_2)$, where $\{e_1,\ldots,e_{d_G(v_j)}\} = E_G(v_j)$ for vertices $v_j \in B$ are added similarly and with a help of additional $3c$ labels.

For $u_i \in R$, to add gadgets $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$, where $\{e_1,\ldots,e_{d_G(u_i)}\} = E_G(u_i)$, we use following additional labels:

- labels $\alpha'_1,\ldots,\alpha'_c$,
- labels $\gamma'_1,\ldots,\gamma'_c$, and
- labels $\gamma''_1,\ldots,\gamma''_c$.

Let us observe that label λ_1 is not used in any join operation in the construction of G'. Hence, it is safe to use it to relabel a vertex as soon as all its incident edges are constructed. Notice also that label λ_3 is not used at all. Thus we can use it to relabel the vertices that should be joined with w_2. We use the working labels ξ_1,ξ_2,ξ_3,ξ_4 as

well.

We scan the expression tree for G' and iteratively change it for each u_i, $i \in \{1,\ldots,n\}$, by adding the corresponding gadgets.

For $i \in \{1,\ldots,n\}$, let $E_G(u_i) = \{e_1,\ldots,e_{d_G(u_i)}\}$. Denote by A_i the set of vertices $\{a_{e_1},\ldots,a_{e_{d_G(u_i)}}\}$ and let $U_i = A_i \cup \{u_i\}$ for $i \in \{1,\ldots,n\}$.

We need the following claim.

Claim 1. *Let X be a node of the expression tree for G' and G'_X be the subgraph of G' corresponding to this node. If $V(G'_X) \cap U_i \neq \emptyset$ but $G'[U_i]$ is not a subgraph of G'_X then the following holds:*

- *if $u_i \in V[G'_X]$, then u_i is labeled by a label which is different from labels of other vertices of G'_X,*
- *the vertices of $A_i \cap V(G'_X)$ that are not adjacent to u_i in G'_X are labeled by labels that are different from the labels of the vertices of $G'_X - A_i$.*

Proof. [Proof of Claim 1] Recall that the vertices $u_1,\ldots,u_n$ are labeled by $\alpha_1,\ldots,\alpha_c$, and these labels are used only for $u_1,\ldots,u_n$ Assume that there are two distinct vertices $u_i,u_j \in V[G'_X]$ labeled by the same label α_p. Because $G'[U_i]$ is not a subgraph of G'_X, there is $a_e \in A_i$ such that $u_ia_e \notin E(G'_X)$. Then a_e should be joined with u_i by an edge on some further step of the construction of G'. But because u_i and u_j have the same label, the join operation that constructs u_ia_e would construct $u_ja_e \notin E(G')$; a contradiction.

Recall that the vertices a_e for $e \in E(G)$ are labeled by $\gamma_1,\ldots,\gamma_c$, and these labels are used only for such vertices. Assume now that there is a vertex of $a_e \in A_i \cap V(G'_X)$ such that $u_ia_e \notin E(G'_X)$ that has the same label as some other vertex $a_{e'} \in V[G'_X]\setminus A_i$. Then again, a_e should be joined with u_i by an edge on some further step of the construction of G'. But because a_e and $a_{e'}$ have the same label, the join operation that constructs u_ia_e would construct $u_ja_{e'}$. Because $a_{e'} \notin A_i$, $u_ja_{e'} \notin E(G')$, and we obtain a contradiction. □

We use Claim 1 to construct graph $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$ in such a way that all z-vertices of this gadget constructed for the node X are labeled by the same label α'_p, if $u_i \in V[G'_X]$ and this vertex is labeled by α_p. If $u_i \notin V[G'_X]$ and the labels $\gamma_{p_1},\ldots,\gamma_{p_h}$ are used for vertices $A_i \cap V(G'_X)$ then all z-vertices are labeled by the labels $\gamma'_{p_1},\ldots,\gamma'_{p_h}$. The construction of the gadget $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$ is completed when after some union operation all vertices of U_i are included in graph G'_X.

When vertex $u_i \in R$ labeled by α_p is introduced, we construct u_i and the columns of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$ that are not associated with $a_{e_1},\ldots,a_{e_{d_G(u_i)}}$ together with joining them edges. To do it, we perform the following set of operations. First, we introduce the vertex u_i labeled by α_p. Then we repeat the following operations $\ell + c(u_i) - d_G(u_i)$ times. A vertex labeled ξ_3 is introduced and then the following operations are repeated $4m$ times:

- introduce a vertex labeled by ξ_1 and $4m+1$ vertices labeled ξ_2,
- join vertices labeled ξ_2 and vertices labeled by ξ_1 and ξ_3,
- relabel vertices labeled by ξ_2 by λ_1, the vertex labeled ξ_3 by ξ_4, and the vertex labeled ξ_1 by ξ_3.

Finally, the vertex labeled by ξ_3 is relabeled by ξ_4, the vertices labeled by ξ_4 are joined with vertices labeled by α'_p (if they exist) and then relabeled by α'_p.

Next we consider the columns associated with w_2 and perform the following $\ell - c(u_i)$ times:

- new vertex labeled by ξ_3 and $4m+1$ vertices labeled by ξ_1 are introduced;

- the vertices labeled by ξ_1 are joined with vertices labeled by ξ_3, and relabeled by λ_3.

Then we repeat the following operations $4m$ times:

- introduce avertex labeled by ξ_1 and $4m+1$ vertices labeled by ξ_2,
- join vertices labeled by ξ_2 and vertices labeled by ξ_1 and ξ_3,
- relabel vertices labeled by ξ_2 by λ_1, the vertex labeled ξ_3 by ξ_4, and the vertex labeled ξ_1 by ξ_3.

Finally, the vertex labeled by ξ_3 is relabeled by ξ_4, the vertices labeled by ξ_4 are joined with vertices labeled α'_p and then relabeled by α'_p.

When a vertex a_e, $u_ia_e \in E(Q)$, labeled by γ_q is introduced, we construct it together with the column of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$ associated with a_e. To do it, we perform the following set of operations. First, we introduce the vertex a_e labeled γ_q and a vertex labeled by ξ_3. Then $4m+1$ vertices labeled ξ_2 are introduced and joined with the vertices labeled γ_p and ξ_3. Then the vertices labeled ξ_2 are relabeled λ_1. Now we repeat the following operations $4m$ times:

- introduce a vertex labeled by ξ_1 and $4m+1$ vertices labeled ξ_2,
- join vertices labeled ξ_2 and vertices labeled ξ_1 and ξ_3,
- relabel vertices labeled ξ_2 by λ_1, the vertex labeled ξ_3 by γ'_q, and the vertex labeled ξ_1 by ξ_3.

Finally, the vertex labeled ξ_3 is relabeled γ'_q.

We are done with introduction nodes. Next we proceed with union operations. Notice that we construct different parts of each gadget $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$ while introducing the vertices of U_i. We join them together when vertices of U_i are collected together by the union operations. Let X be a union node of the expression tree for G'. Denote by X and Y two children of this node and let G'_Y and G'_Z be the subgraphs of G' corresponding to these nodes. Denote also by Q_Y and Q_Z the subgraph of Q constructed for Y and Z respectively. Assume inductively, that for every $i\in\{1,\ldots,n\}$,

- if $u_i \in V(G'_Y)$ ($u_i \in V(G'_Z)$ respectively) and u_i is labeled by α_p, then the z-vertices of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$ that are in Q_Y (in Q_Z respectively) are labeled by α'_p.
- if $u_i \notin V(G'_Y)$ ($u_i \notin V(G'_Z)$ respectively) and $a_e \in A_i$ is labeled by γ_p in G'_Y (in G'_Z respectively), then the z-vertices of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$ that are in the column associated with a_e are labeled by γ'_p in Q_Y (Q_Z respectively).
- all z-vertices of distinct columns $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$ in Q_Y (Q_Z respectively) are pairwise adjacent.
- $\gamma''_1,\ldots,\gamma''_c$ are not used in Q_Y and Q_Z.

It is straightforward to see that these conditions hold if Y (Z respectively) is an introduce node.

If for every $i\in\{1,\ldots,n\}$, G'_Y or G'_Z has no vertices of U_i, then we just perform the union operation. Otherwise, we first relabel γ'_p to γ''_p for $p\in\{1,\ldots,c\}$ in Q_Y. Then we perform the union operation. Now we consider $i\in\{1,\ldots,n\}$ such that $U_i\cap V(G'_Y)\neq\emptyset$ and $U_i\cap V(G'_Z)\neq\emptyset$. We have three cases.

Case 1. $u_i\in V(G'_Y)$. Suppose that u_i is labeled by α_p. By Claim 1, α_p is used only for u_i in G'_X. Hence, α'_p is used only for z-vertices of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$ in Q_Y. Suppose that graph G'_Z includes vertices of A_i labeled by $\gamma_{p_1},\ldots,\gamma_{p_h}$. Then all z-vertices of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1},\ldots,a_{e_{d_G(u_i)}},w_2)$ in the columns associated with a vertex labeled by γ_{p_j} are labeled by γ'_{p_j} for $j\in\{1,\ldots,h\}$ in Q_Z. Because $u_i\notin V(G'_Z)$,

the vertices of $A_i \cap V(G'_Z)$ are not adjacent to u_i in G'_X. By Claim 1, labels $\gamma_{p_1}, \ldots, \gamma_{p_h}$ are not used for $V(G'_X) \setminus A_i$. Therefore, labels $\gamma'_{p_1}, \ldots, \gamma'_{p_h}$ are used only for z-vertices of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1}, \ldots, a_{e_{d_G(u_i)}}, w_2)$ in the columns associated with the vertices labeled by $\gamma_{p_1}, \ldots, \gamma_{p_h}$. For $j \in \{1, \ldots, h\}$, we join the vertices labeled by α'_p and γ'_{p_j}, and then relabel the vertices labeled γ'_{p_j} by α'_p. It remains to observe that in this way we join the z-verties of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1}, \ldots, a_{e_{d_G(u_i)}}, w_2)$ from G'_Y and G'_Z, and afterwards all these vertices are labeled by α'_p.

Case 2. $u_i \in V(G'_Z)$. This case is symmetric to Case 1, and we use the same arguments. Suppose that u_i is labeled by α_p. Then α'_p is used only for z-veritices of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1}, \ldots, a_{e_{d_G(u_i)}}, w_2)$ in Q_Z. Suppose that the graph G'_Y includes vertices of A_i labeled by labels $\gamma_{p_1}, \ldots, \gamma_{p_h}$. Then all z-vertices of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1}, \ldots, a_{e_{d_G(u_i)}}, w_2)$ in the columns associated with a vertex labeled by γ_{p_j} are labeled by γ''_{p_j} for $j \in \{1, \ldots, h\}$. For $j \in \{1, \ldots, h\}$, we join the vertices labeled α'_p and γ''_{p_j}, an then relabel the vertices labeled γ''_{p_j} by α'_p.

Case 3. $u_i \notin V(G'_Y) \cup V(G'_Z)$. Suppose that G'_Y includes vertices of A_i labeled by $\gamma_{p_1}, \ldots, \gamma_{p_h}$, and G'_Z has vertices of A_i labeled by $\gamma_{q_1}, \ldots, \gamma_{q_f}$. Then all z-vertices of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1}, \ldots, a_{e_{d_G(u_i)}}, w_2)$ in the columns associated with a vertex labeled by γ_{p_j} in G'_Y are labeled by γ''_{p_j} for $j \in \{1, \ldots, h\}$, and all z-vertices of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1}, \ldots, a_{e_{d_G(u_i)}}, w_2)$ in the columns associated with a vertex labeled by γ_{q_j} in G'_Z are labeled by γ'_{q_j} for $j \in \{1, \ldots, f\}$. By Claim 1, $\gamma''_{p_1}, \ldots, \gamma''_{p_h}$ and $\gamma'_{q_1}, \ldots, \gamma'_{q_f}$ are used only for the z-vertices of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1}, \ldots, a_{e_{d_G(u_i)}}, w_2)$. For each $j \in \{1, \ldots, h\}$ and each $j' \in \{1, \ldots, f\}$, we join the vertices labeled by γ''_{p_j} and $\gamma'_{q_{j'}}$. Clearly, in this way we join the z-verties of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1}, \ldots, a_{e_{d_G(u_i)}}, w_2)$ from Q_Y and Q_Z.

In conclusion we relabel γ''_p to γ'_p for $p \in \{1, \ldots, c\}$.

Observe that for the subgraph of Q_X constructed for X, we have the required properties. Namely, if $u_i \in V(G'_X)$ and u_i is labeled by α_p, then the z-vertices of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1}, \ldots, a_{e_{d_G(u_i)}}, w_2)$ that are in Q_X are labeled α'_p, and if $u_i \notin V(G'_X)$ and $a_e \in A_i$ is labeled by γ_p in G'_X, then the z-vertices of $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1}, \ldots, a_{e_{d_G(u_i)}}, w_2)$ that are in the column associated with a_e are labeled by γ'_p. Also all z-vertices of distinct columns $H_{d_G(u_i),\ell-c(u_i)}(a_{e_1}, \ldots, a_{e_{d_G(u_i)}}, w_2)$ in Q_X are pairwise adjacent, and $\gamma''_1, \ldots, \gamma''_c$ are not used in Q_X.

The join operations in the expression tree for G' are done in the new tree in exactly the same way. The relabel operation in the expression tree of G', that is, relabel α_p to α_q and relabel γ_p to γ_q, are replaced by relabel α_p to α_q, α'_p to α'_q, and γ_p to γ_q, γ'_p to γ'_q, respectively. Notice that this relabeling does not violate the aforementioned requirements for labelings that are crucial for the union operations.

When we have completed the scan of the expression tree for G', the only thing which remains is to join vertices labeled λ_3 and the vertex labeled ζ_2 (the vertex w_2).

Gadgets $H_{d_G(v_j),\ell-1}(a_{e_1}, \ldots, a_{e_{d_G(v_j)}}, w_2)$ for vertices $v_j \in B$, where $\{e_1, \ldots, e_{d_G(v_j)}\} = E_G(v_j)$, are added in the same way by using additional $3c$ labels and labels $\lambda_1, \lambda_3, \xi_1, \xi_2, \xi_3, \xi_4$. Observe also that Claim 1 can be reformulated for each v_j and $E_G(v_j)$. This completes the proof of Lemma 4.6. □

To conclude the first part of the proof of the Theorem 4.1, we observe that the number of vertices of Q is polynomial in $n + r$, and therefore if Max-Cut is solvable in time $f(\mathbf{cwd}(Q)) \cdot |V(Q)|^{o(\mathbf{cwd}(Q))}$ then Red-Blue CDS is solvable in time $f(\mathbf{fvs}(G)) \cdot |V(G)|^{o(\mathbf{fvs}(G))}$.

4.2. Algorithmic upper bound for Max-Cut.. Now we outline an algorithm for solving Max-Cut in time $n^{O(t)}$ on graphs given together with an expression tree of width at most t. The algorithm is based on dynamic programming over the expression tree of the input graph. We first describe what we store in the tables corresponding to the nodes in the expression tree.

Let G be a graph with n vertices and m edges, and let T be an expression tree for G of width t. By the results of Courcelle and Olariu [6], without loss of generality we can assume that T is irredundant. For a node X of T, denote by G_X the t-graph associated with this node, and let $U_1(X),\ldots,U_t(X)$ be the sets of vertices of G_X labeled $1,\ldots,t$ respectively. The table for the node X stores vectors $(s_1,\ldots,s_t,r)$ of integers such that $0 \leq s_i \leq |U_i(X)|$ for $1 \leq i \leq t$, and $0 \leq r \leq |E(G_X)|$, for which there is a partition V_1, V_2 of $V(G_X)$ such that $|V_1 \cap U_i(G_X)| = s_i$ and $|C_{G_X}(V_1,V_2)| \geq r$. Notice that this table contains at most $(n+1)^t \cdot m$ vectors. If X is the root node of T (that is, $G = G_X$) then $\mathbf{mcut}(G)$ is equal to the maximum value of r for which the table for X contains an entry with this value.

Now we provide the details of how to construct and update such tables. The construction for introduce nodes of T is straightforward.

Relabel Node: Suppose that X is a relabel node $\rho_{i\to j}$, and let Y be the child of X. Then the table for X contains a vector $(s_1,\ldots,s_t,r)$ if and only if $s_i = 0$ and the table for Y contains the entry $(s'_1,\ldots,s'_t,r)$ such that $s'_p = s_p$ for $1 \leq p \leq t$, $t \neq i,j$, and $s_j = s'_i + s'_j$.

Union Node: Let X be a union node with children Y and Z. In this case the table for X contains vector $(s_1,\ldots,s_t,r)$ if and only if the tables for Y and Z have vectors $(s'_1,\ldots,s'_t,r')$ and $(s''_1,\ldots,s''_t,r'')$ respectively, such that $s'_i + s''_i = s_i$ for $1 \leq i \leq t$, and $r' + r'' \geq r$.

Join Node: Finally, suppose that X is a join node $\eta_{i,j}$ with the child Y. The table for X has vector $(s_1,\ldots,s_t,r)$ if and only if the table for Y includes a vector $(s_1,\ldots,s_t,r')$ such that $r' + s_i(|U_j(Y)| - s_j) + s_j(|U_i(Y)| - s_i) \geq r$.

Correctness of the algorithm follows from the description of the procedure. The running time of the algorithm is $O(t^{O(1)}n^{2t+O(1)})$. This proves that Max-Cut can be solved in time $n^{O(t)}$ if a graph with a clique decomposition of width at most t is given.

4.3. Bipartization by Edge Removal and Maximum (Minimum) Bisection.. Theorem 4.1 have several interesting corollaries for similar problems like Bipartization by Edge Removal and Maximum (Minimum) Bisection.

In the Bipartization by Edge Removal problem, we are given a graph G and a positive integer k, and the question is whether there is a set of edges X such that $|X| \leq k$ and the graph G' with the vertex set $V(G)$ and the edge set $E(G) \setminus X$ is bipartite. Since this problem is dual to the Maximum Cut problem, we immediately have the following corollary.

Corollary 4.7. *Let G be an n-vertex graph given together with an expression tree of width t. Then the* Bipartization by Edge Removal *problem*

- *cannot be solved in time $f(t)\cdot n^{o(t)}$ unless the ETH fails;*
- *is solvable in time $n^{O(t)}$.*

In the Maximum (Minimum) Bisection problem, we are given a graph G with an even number of vertices and a positive integer k, and the objective is to check whether there is a partition of $V(G)$ into two sets V_1 and V_2 of equal size such that $|C_G(V_1,V_2)| \geq k$ ($|C_G(V_1,V_2)| \leq k$).

Corollary 4.8. *Let G be an n-vertex graph given together with an expression*

tree of width t. Then the Maximum (Minimum) Bisection *problem*

- *cannot be solved in time $f(t) \cdot n^{o(t)}$ unless the ETH fails;*
- *is solvable in time $n^{O(t)}$.*

Proof. The algorithmic upper bounds for Maximum Bisection and Minimum Bisection follow from an easy modification of the algorithm for Max-Cut described in Section 4.2. The lower bound can be obtained from the fact that the Max-Cut problem for a graph G can be reduced to Maximum Bisection by adding $|V(G)|$ isolated vertices. The claim about the Minimum Bisection follows from the observation that the Maximum Bisection problem for a graph G can be reduced to Minimum Bisection for the complement graph $\overline{G}$, and the fact that $\mathbf{cwd}(\overline{G}) \leq 2 \cdot \mathbf{cwd}(G)$ (see [35, 6]). □

5. Edge dominating set. In this section, we consider the Edge Dominating Set problem. In the Edge Dominating Set problem, we are given a graph G and a positive integer k, and the objective is to determine whether there is a set of edges $X \subseteq E(G)$ such that $|X| \leq k$ and every edge of G is either included in X, or it is adjacent to at least one edge of X (which *dominates* it). The set X is called an *edge dominating set* of G. We prove the following result for Edge Dominating Set.

Theorem 5.1. *Let G be an n-vertex graph given together with an expression tree of width t. Then the* Edge Dominating Set *problem*

- *cannot be solved in time $f(t) \cdot n^{o(t)}$ unless the ETH fails;*
- *is solvable in time $n^{O(t)}$.*

The remaining part of this section is devoted to the proof of Theorem 5.1. We first show the lower bound and then complement this result with the corresponding upper bound.

5.1. Lower bound. To prove our result we give a linear FPT-reduction from Red-Blue Exact Saturated CDS to Edge Dominating Set. The proof is organized as follows: we first give a construction, then prove its correctness and finally argue on the clique-width of the transformed instance. We start with descriptions of auxiliary gadgets.

Auxiliary gadgets.. Let $s \leq t$ be positive integers. We construct graph $F_{s,t}$ with the vertex set $\{x_1, \ldots, x_s, y_1, \ldots, y_s, z_1, \ldots, z_t\}$ 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 a complete bipartite graph between y_i's and z_j's with pendent vertices attached to y_i's. The vertices $z_1, z_2, \ldots, z_t$ are called *roots* of $F_{s,t}$. Further we refer to the vertices x_i, y_j as *x and y-vertices* of the gadget.

Graph $F_{s,t}$ has the following property.

Lemma 5.2. *Any set of s edges incident with vertices $y_1, \ldots, y_s$ forms an edge dominating set in $F_{s,t}$. Furthermore, let G be a graph obtained by taking the union of $F_{s,t}$ with some other graph H such that $V(F_{s,t}) \cap V(H) = \{z_1, \ldots, z_t\}$. 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 $i \in \{1, \ldots, s\}$.

Final reduction.. Now we describe our reduction. Let (G, c) be a red-blue capacitated graph with $R = \{u_1, \ldots, u_n\}$ being the set of red vertices and $B = \{v_1, \ldots, v_r\}$ being the set of blue vertices, and let k be a positive integer. Each red vertex u_i is replaced by the set U_i with $c(u_i)$ vertices, and for every edge $u_i v_j \in E(G)$, all vertices of U_i are joined to v_j by edges. For every vertex v_j we add one additional leaf (a pendent vertex). Now vertex sets $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_n\}$ are constructed, and vertices a_i are made adjacent to all the vertices of U_i and the vertex b_i. For every

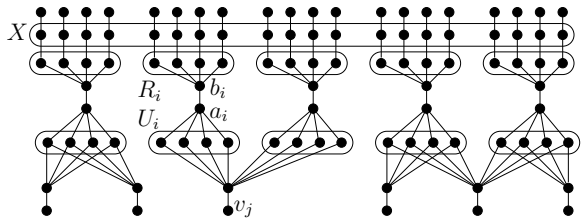

FIG. 3. *Graph G'*

vertex b_i, a set R_i of $c(u_i)$ vertices is added and b_i is made adjacent to all the vertices in R_i. Then to every vertex of $R_1 \cup R_2 \cup \cdots \cup R_n$ we add 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 3). Finally, we introduce three copies of $F_{s,t}$:

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

Let this final resulting graph be H.

LEMMA 5.3. *A graph G has a saturated capacitated dominating set of size k if and only if H has an edge dominating set of cardinality at most $n + r + \ell$.*

Proof. Let S be an exact saturated dominating set of size k in G and f be its corresponding domination mapping. For convenience, we assume that $S = \{u_1, \ldots, u_k\}$. We construct an edge dominating set as follows. First we select an edge emanating from every vertex in the set $\{v_1, \ldots, v_r\}$. For every vertex v_j, we choose a vertex u in U_i where $u_i = f(v_j)$ 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 dominate all the edges in the sets $E(v_j)$, $1 \leq j \leq r$, and in sets $E(u)$ for $u \in U_1 \cup \cdots \cup U_k$. Now we add $n - k$ edges from $F_{n-k,n}$ which are incident with vertices in $\{a_{k+1}, \ldots, a_n\}$ and k edges from $F_{k,n}$ which are incident to $\{b_1, \ldots, b_k\}$. Then $\ell - r$ matching edges joining vertices of $R_{k+1}, \ldots, R_n$ to the vertices of X are included in the set. Finally, we add r edges form $F_{r,\ell}$ which are incident to vertices of X and are adjacent to vertices of $R_1, \ldots, R_k$. Since S is an exact capacitated dominating set, $\sum_{i=1}^{k} c(u_i) = r$, and from our description it is clear that the resulting set is an edge dominating set of size $n + r + \ell$ for H.

We proceed to prove the other direction of the equivalence. Let L be an edge dominating set of cardinality at most $n + r + \ell$. The set L is forced to contain at least one edge from every $E(v_j)$, 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 the presence of pendent edges. This implies that $|L| = n + r + \ell$, 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 $a_i b_i$ has 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 $I = \{i : a_i \text{ is incident with an edge from } L \cap E(F_{n-k,n})\}$ 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 $\{1, \ldots, n\}$. 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 r edges of $F_{r,\ell}$ are included in L, we have that

$\sum_{i\in I}|R_i| \leq \ell - r$ and

$$\sum_{j\in J}|R_j| = r - \sum_{i\in I}|R_i| \geq \ell - (\ell - r) \geq r.$$

Let $S = \{u_i \colon i \in J\}$. Clearly, $|S| = k$. Now we show that S is a saturated capacitated dominating set. For $i \in J$, edges which join a vertex a_i to U_i are not dominated by edges from $L \cap E(F_{n-k,k})$, and hence they have to be dominated by edges from sets $E(v_j)$. Since $r \leq \sum_{i\in J}|R_j| = \sum_{i\in J}|U_j|$, there are exactly r such edges, and every such edge must be dominated by exactly one edge from L. We also know that $L \cap E(v_j) \neq \emptyset$ for all $j \in \{1,\ldots,r\}$ and hence for every v_j, there is exactly one edge which joins it with some vertex $u \in U_i$ for some $i \in J$. Furthermore, all these edges are not adjacent, that is, they form a matching. We define $f(v_j) = u_i$ for $j \in \{1,\ldots,r\}$. 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 feedback vertex number and has the properties required by Lemma 2.1, then H has bounded clique-width.

LEMMA 5.4. *If for every minimum feedback vertex set X of G, each vertex v of the forest $G-X$ is adjacent with at most one vertex of X, then* $\mathbf{cwd}(H) \leq 4\cdot\mathbf{fvs}(G)+13$.

Proof. By Lemma 2.1, we have that the graph G is of clique-width at most $s = 4\cdot\mathbf{fvs}(G)+3$. Suppose that the expression tree for G uses s-labels $\{\alpha_1,\ldots,\alpha_s\}$. We construct the expression tree for H by scanning the expression tree for G. To construct the expression tree for H, we need the following additional labels:

- Labels ξ_1, ξ_2, and ξ_3 for attaching $F_{n-k,n}$, $F_{k,n}$ and $F_{r,\ell}$ respectively.
- Label λ for marking some vertices that are already joined with all the neighbors.
- Working labels $\gamma_1,\ldots,\gamma_6$.

When a vertex $u_i \in R$ labeled by α_p is introduced, we perform the following set of operations. First we introduce the following vertices with working labels: $c(u_i)$ vertices of U_i with label γ_1, the vertex a_i with label γ_2, and the vertex b_i with label γ_3. Then we join the vertices labeled by γ_1 with the vertex labeled with γ_2, and the vertex labeled by γ_2 with the vertex labeled γ_3. Then we relabel γ_1 to α_p and γ_2 to ξ_1. Now we create vertices of R_i and the paths attached to it. To do so we perform the following operations $c(u_i)$ times:

- introduce three nodes labeled by γ_4, γ_5 and γ_6
- join γ_3 with γ_4, γ_4 with γ_5 and γ_5 with γ_6,
- relabel γ_4 to λ, γ_5 to ξ_3, and γ_6 to λ.

Finally, we relabel γ_3 to ξ_2. We omit the union operations from the description and assume that if some vertex is introduced then this operation is immediately performed.

If a vertex $v_j \in B$ labeled by α_q is introduced, then we introduce vertex v_i labeled by γ_1, and a pendent vertex labeled by γ_2, join vertices labeled by γ_1 and γ_2, and then relabel γ_1 to α_q and γ_2 to λ.

If in the expression tree of G, the join operation occurs between two labels, say α_p and α_q, then we repeat in the new tree. Union operations in the expression tree are done exactly as before.

Finally, to complete the construction of the expression tree for H, we add $F_{n-k,n}$, $F_{k,n}$ and $F_{r,\ell}$. Notice that all the vertices in $\{a_1,\ldots,a_n\}$, $\{b_1,\ldots,b_n\}$ and X are labeled by ξ_1, ξ_2 and ξ_3 respectively. From here we can easily add $F_{n-k,n}$, $F_{k,n}$ and $F_{r,\ell}$ with root vertices $\{a_1,\ldots,a_n\}$, $\{b_1,\ldots,b_n\}$ and X respectively by making use of

working labels $\gamma_1, \gamma_2, \gamma_3$, and λ. To obtain each gadget, we construct x and y-vertices labeled by γ_1 and γ_2 respectively, join γ_1 with γ_2, and then relabel γ_1 to λ and γ_2 to γ_3. As soon as x- and y-vertices of each gadget are constructed and joined by a matching, we join γ_3 with ξ_1, ξ_2 or ξ_3 to construct t $F_{n-k,n}$, $F_{k,n}$, or $F_{r,\ell}$ respectively. When the construction of each gadget is completed, we relabel γ_3 to λ. This concludes the description for the expression tree for H. □

To conclude the lower bound proof of Theorem 5.1, it remains to note that H has $4(n+r+\ell) \leq 4(n+r+n^2)$ vertices, and therefore if we could solve Edge Dominating Set in time $f(t)|V(H)|^{o(t)}$, where $t = \mathbf{cwd}(H)$, then we also would be able to solve Red-Blue Exact Saturated CDS in time $f(k)|V(G)|^{o(k)}$, where k is the feedback vertex set number of G. By Theorem 3.1, this would imply that the ETH does not hold.

5.2. Algorithmic upper bound for Edge Dominating Set. Now we give an algorithmic upper bound for the Edge Dominating Set problem parameterized by the clique-width, that is, give an algorithm running in time $n^{O(t)}$ for Edge Dominating Set on graphs with a given expression tree of with at most t. Again, the algorithm is based on a dynamic programming over the expression tree of the input graph.

Let G be a graph with n vertices and m edges, and let T be an expression tree for G of width t. We assume that T is irredundant. For a node X of T, denote by G_X the t-graph associated with this node, and let $U_1(X), \ldots, U_t(X)$ be sets of vertices of G_X labeled $1, \ldots, t$ respectively. The table of data for the node X stores vectors $(s_1, \ldots, s_t, r_1, \ldots, r_t, l)$ of non-negative integers with the following properties:

- $s_i + r_i \leq |U_i(X)|$ for $1 \leq i \leq t$;
- $l \leq |E(G_X)|$;
- there is a set of edges $S \subseteq E(G_X)$ such that s_i vertices of $U_i(X)$ are adjacent to the edges of S for $1 \leq i \leq t$, and $|S| \leq l$;
- it is possible to attach r_i pendent edges to the vertices of $U_i(X)$ for $1 \leq i \leq t$ in such a way that these edges dominate all edges of G_X undominated by S.

The intuition behind this definition is that there are at most r_i vertices in $U_i(X)$ that are incident with at least one edge $e \notin S$ in G_X that is not dominated by S. The size of this table is at most $(n+1)^{2t} \cdot m$. If X is the root node of T (that is $G = G_X$) then the size of the minimum edge dominating set is the minimum value of l for which the table for X contains an entry with the value of the parameter being l and $r_1 = \ldots = r_t = 0$.

Now we give the details of how we make our tables and how do we update it.

Introduce Node: Tables for introduce nodes of T are constructed in a straightforward way.

Relabel Node: Let X be a relabel node $\rho_{i \to j}$, and let Y be the child of X. Then the table for X contains a vector $(s_1, \ldots, s_t, r_1, \ldots, r_s, l)$ if and only if $s_i = 0$, $r_i = 0$ and the table for Y contains the entry $(s'_1, \ldots, s'_t, r'_1, \ldots, r'_t, l)$ such that $s'_p = s_p$ and $r_p = r'_p$ for $1 \leq p \leq t$, $t \neq i, j$, and $s_j = s'_i + s'_j$, $r_j = r'_i + r'_j$.

Union Node: Let X be a union node with children Y and Z. In this case the table for X contains a vector $(s_1, \ldots, s_t, r_1, \ldots, r_t, l)$ if and only if the tables for Y and Z have vectors $(s'_1, \ldots, s'_t, r'_1, \ldots, r'_t, l')$ and $(s''_1, \ldots, s''_t, r''_1, \ldots, r''_t, l'')$ respectively such that $s'_i + s''_i = s_i$ and $r'_i + r''_i = r_i$ for $1 \leq i \leq t$, and $l' + l'' \leq l$.

Join Node: Finally, suppose that X is a join node $\eta_{i,j}$ with the child Y. It can be noted that the table for X has a vector $(s_1, \ldots, s_t, r_1, \ldots, r_t, l)$ if and only if

the table for Y includes a vector $(s'_1, \ldots, s'_t, r'_1, \ldots, r'_t, l')$ such that

- $s_p = s'_p$ and $r_p = r'_p$ for $1 \leq p \leq t$, $p \neq i, j$;
- $s_i + r_i = s'_i + r'_i$, $s_j + r_j = s'_j + r'_j$;
- $s'_i \leq s_i$, $s'_j \leq s_j$;
- either $s'_i + r'_i = |U_i(Y)|$, or $s'_j + r'_j = |U_j(Y)|$;
- $l' + \max\{s_i - s'_i, s_j - s'_j\} \leq l$.

Correctness of the algorithm follows from the description of the algorithm, and its running time $O(t^{O(1)}n^{4t+O(1)})$ is. Hence Edge Dominating Set is solvable in time $n^{O(t)}$. This concludes the proof of Theorem 5.1.

6. Conclusion and further directions. In this paper, we obtained the first asymptotically tight bounds for problems parameterized by the clique-width of the input graph. In particular, we showed that Max-Cut and Edge Dominating Set cannot be solved in time $f(t)n^{o(t)}$ unless the ETH collapses; while there do exist algorithms with running time $n^{O(t)}$ for both these problems, where t is the clique-width of the input graph. Notice that our reduction to obtain a tight lower bound for Max-Cut is also an FPT-reduction and, therefore, Max-Cut is W[1]-hard when parameterized by the clique-width of the input graph. Thus we resolve an open problem about the parameterized complexity of Max-Cut.

We conclude with an open problem related to Hamiltonian Cycle. In the Hamiltonian Cycle problem, we are given a graph G and the objective is to check whether there exists a cycle passing through every vertex of G. Similar to Max-Cut and Edge Dominating Set we can obtain the following algorithmic lower bound for the Hamiltonian Cycle problem when parameterized by the clique-width of the input graph.

Theorem 6.1. *Assuming ETH, the* Hamiltonian Cycle *problem cannot be solved in time $f(t)n^{o(t)}$, where n is the number of vertices and t is the clique-width of the input graph.*

However, all the algorithms we know for Hamiltonian Cycle run in time $n^{O(t^2)}$ if an expression tree of width t is given. We leave it open to find either an improved lower bound or an improved upper bound for the Hamiltonian Cycle problem.

REFERENCES

[1] H. Bodlaender, D. Lokshtanov, and E.Penninkx, *Planar capacitated dominating set is W[1]-hard*, in IWPEC'09, vol. 5917 of LNCS, Springer, 2009, pp. 50–60.

[2] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia, *On the computational hardness based on linear FPT-reductions*, J. Comb. Optim., 11 (2006), pp. 231–247.

[3] ———, *Strong computational lower bounds via parameterized complexity*, J. Comput. Syst. Sci., 72 (2006), pp. 1346–1367.

[4] Derek G. Corneil, Michel Habib, Jean-Marc Lanlignel, Bruce A. Reed, and Udi Rotics, *Polynomial time recognition of clique-width $\leq$ 3 graphs (extended abstract)*, in LATIN'00, vol. 1776 of LNCS, Springer, 2000, pp. 126–134.

[5] 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.

[6] Bruno Courcelle and Stephan Olariu, *Upper bounds to the clique width of graphs*, Discrete Appl. Math., 101 (2000), pp. 77–114.

[7] Michael Dom, Daniel Lokshtanov, Saket Saurabh, and Yngve Villanger, *Capacitated domination and covering: A parameterized perspective*, in IWPEC'08, vol. 5018 of LNCS, Springer, 2008, pp. 78–90.

[8] Rodney G. Downey, Vladimir Estivill-Castro, Michael R. Fellows, Elena Prieto, and Frances A. Rosamond, *Cutting up is hard to do: the parameterized complexity of k-cut and related problems*, Electr. Notes Theor. Comput. Sci., 78 (2003).

[9] Rodney G. Downey and Michael R. Fellows, *Fundamentals of Parameterized Complexity*, Texts in Computer Science, Springer, 2013.
[10] Wolfgang Espelage, Frank Gurski, and Egon 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.
[11] Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette, *On the parameterized complexity of multiple-interval graph problems*, Theor. Comput. Sci., 410 (2009), pp. 53–61.
[12] Michael R. Fellows, Frances A. Rosamond, Udi Rotics, and Stefan Szeider, *Clique-width is NP-complete*, SIAM J. Discrete Math., 23 (2009), pp. 909–939.
[13] J. Flum and M. Grohe, *Parameterized complexity theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.
[14] Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh, *Algorithmic lower bounds for problems parameterized by clique-width*, in SODA, 2010, pp. 493–502.
[15] Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh, *Intractability of clique-width parameterizations*, SIAM J. Comput., 39 (2010), pp. 1941–1956.
[16] Michael U. Gerber and Daniel Kobler, *Algorithms for vertex-partitioning problems on graphs with fixed clique-width*, Theoret. Comput. Sci., 299 (2003), pp. 719–734.
[17] Omer Giménez, Petr Hlinený, and Marc Noy, *Computing the tutte polynomial on graphs of bounded clique-width*, SIAM J. Discret. Math., 20 (2006).
[18] Benny Godlin, Tomer Kotek, and Johann A. Makowsky, *Evaluations of graph polynomials*, in WG, vol. 5344 of Lecture Notes in Computer Science, 2008, pp. 183–194.
[19] Petr Hlinený and Sang il Oum, *Finding branch-decompositions and rank-decompositions*, SIAM J. Comput., 38 (2008), pp. 1012–1032.
[20] Petr Hlinený, S.-il Oum, Detlef Seese, and Georg Gottlob, *Width parameters beyond tree-width and their applications*, Comput. J., 51 (2008), pp. 326–362.
[21] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane, *Which problems have strongly exponential complexity?*, J. Comput. Syst. Sci., 63 (2001), pp. 512–530.
[22] ———, *Which problems have strongly exponential complexity?*, J. Comput. Syst. Sci., 63 (2001), pp. 512–530.
[23] Klaus Jansen, Stefan Kratsch, Dániel Marx, and Ildikó Schlotter, *Bin packing with fixed number of bins revisited*, in SWAT, vol. 6139 of LNCS, Springer, 2010, pp. 260–272.
[24] Marcin Kaminski, Vadim V. Lozin, and Martin Milanic, *Recent developments on graphs of bounded clique-width*, Discrete Applied Mathematics, 157 (2009), pp. 2747–2761.
[25] Daniel Kobler and Udi Rotics, *Polynomial algorithms for partitioning problems on graphs with fixed clique-width (extended abstract)*, in SODA'01, ACM-SIAM, 2001, pp. 468–476.
[26] ———, *Edge dominating set and colorings on graphs with fixed clique-width*, Discrete Appl. Math., 126 (2003), pp. 197–221.
[27] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh, *Lower bounds based on the exponential time hypothesis*, Bulletin of the EATCS, 105 (2011), pp. 41–72.
[28] J.A. 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.
[29] Dániel Marx, *On the optimality of planar and geometric approximation schemes*, in FOCS, 2007, pp. 338–348.
[30] ———, *Can you beat treewidth?*, Theory of Computing, 6 (2010), pp. 85–112.
[31] S.-il Oum and Paul Seymour, *Approximating clique-width and branch-width*, J. Combin. Theory Ser. B, 96 (2006), pp. 514–528.
[32] Rolf Niedermeier, *Invitation to fixed-parameter algorithms*, vol. 31 of Oxford Lecture Series in Mathematics and its Applications, Oxford University Press, Oxford, 2006.
[33] Michaël Rao, *MSOL partitioning problems on graphs of bounded treewidth and clique-width*, Theoret. Comput. Sci., 377 (2007), pp. 260–267.
[34] Karol Suchan and Ioan Todinca, *On powers of graphs of bounded NLC-width (clique-width)*, Discrete Appl. Math., 155 (2007), pp. 1885–1893.
[35] Egon Wanke, *k-NLC graphs and polynomial algorithms*, Discrete Appl. Math., 54 (1994), pp. 251–266.