Tight Complexity Bounds for FPT Subgraph Problems Parameterized by Clique-Width*

Hajo Broersma, Petr A. Golovach, and Viresh Patel

School of Engineering and Computing Sciences, Durham University,
Science Laboratories, South Road, Durham DH1 3LE, UK
{hajo.broersma,petr.golovach,viresh.patel}@durham.ac.uk

Abstract. We give tight algorithmic lower and upper bounds for some double-parameterized subgraph problems when the clique-width of the input graph is one of the parameters. Let G be an arbitrary input graph on n vertices with clique-width at most w. We prove the following results.

- The Dense (Sparse) k-Subgraph problem, which asks whether there exists an induced subgraph of G with k vertices and at least q edges (at most q edges, respectively), can be solved in time $k^{O(w)} \cdot n$, but it cannot be solved in time $2^{o(w \log k)} \cdot n^{O(1)}$ unless the Exponential Time Hypothesis (ETH) fails.
- The d-Regular Induced Subgraph problem, which asks whether there exists a d-regular induced subgraph of G, and the Minimum Subgraph of Minimum Degree at least d problem, which asks whether there exists a subgraph of G with k vertices and minimum degree at least d, can be solved in time $d^{O(w)} \cdot n$, but they cannot be solved in time $2^{o(w \log d)} \cdot n^{O(1)}$ unless ETH fails.

1 Introduction

The notion of clique-width introduced by Courcelle and Olariu [14] (we refer the reader to the survey [24] for further information on different width parameters) has now become one of the fundamental parameters in Graph Algorithms. Many problems which are hard on general graphs can be solved efficiently when the input is restricted to graphs of bounded clique-width. The meta-theorem of Courcelle, Makowsky, and Rotics [13] states that all problems expressible in MS_1-logic are *fixed parameter tractable* (FPT), when parameterized by the clique-width of the input graph (see the books of Downey and Fellows [18] and Flum and Grohe [21] for a detailed treatment of parameterized complexity). In other words, this theorem shows that any problem expressible in MS_1-logic can be solved for graphs of clique-width at most w in time $f(w) \cdot |I|^{O(1)}$, where $|I|$ is the size of the input and f is a computable function depending on the parameter w only. Here, the superexponential function f is defined by a logic formula, and it grows very fast.

The basic method for constructing algorithms for graphs of bounded clique-width is to use dynamic programming along an expression tree (the definition

* This work is supported by EPSRC (EP/G043434/1 and F064551/1).

D. Marx and P. Rossmanith (Eds.): IPEC 2011, LNCS 7112, pp. 207–218, 2012.

is given in Section 2). Computing clique-width is an NP-hard problem [20], but it can be approximated and a corresponding expression tree can be constructed in FPT-time [23,30]. In our paper it is always assumed that an expression tree is given. In this case dynamic programming algorithms can be relatively efficient: usually single-exponential in the clique-width. A natural question to ask is whether the running times of such algorithms are asymptotically optimal up to some reasonable complexity conjectures.

The Exponential Time Hypothesis has proved to be an effective tool for establishing tight complexity bounds for parameterized problems, but there are still not many results of this nature in the literature. The *Exponential Time Hypothesis (ETH)* [25] asserts that there does not exist an algorithm for solving 3-SAT running in time $2^{o(n)}$ on a formula with n variables; this is equivalent to the parameterized complexity conjecture that FPT $\neq$ M[1] [17,21]. Chen et al. [8,9,10] showed that there is no algorithm for k-Clique running in time $f(k)n^{o(k)}$, for n-vertex graphs, unless ETH fails (on the other hand it is easily seen that k-Clique can be solved in time $n^{O(k)}$). The lower bound on the k-Clique problem can be extended to some other parameterized problems via linear FPT-reductions [9,10]. In particular, for problems parameterized by clique-width, Fomin et al. [22] proved that Max-Cut and Edge Dominating Set cannot be solved in time $f(w)n^{o(w)}$ on n-vertex graphs of clique-width at most w, unless ETH collapses. For FPT problems, Cai and Juedes [6] proved that the parameterized version of any MaxSNP-complete problem cannot be solved in time $2^{o(k)} \cdot |I|^{O(1)}$ if ETH holds. Here k is the natural parameter of an MaxSNP-complete problem with the instance I, i.e. the maximized function should have a value at least k.

Lokshtanov, Marx and Saurabh [28] considered several FPT problems solvable in time $2^{O(k \log k)} \cdot |I|^{O(1)}$ and showed that a $2^{o(k \log k)} \cdot |I|^{O(1)}$-time algorithm for these problems would violate ETH. To do this, they introduced special restricted versions of some basic problems like k-Clique on graphs with k^2 vertices (and with some other restrictions) and proved that these problems cannot be solved in time $2^{o(k \log k)} \cdot k^{O(1)}$ unless ETH collapses. These results open the possibility of establishing algorithmic lower bounds for natural problems. We use this approach to prove asymptotically tight bounds for some double-parameterized subgraph problems when the clique-width of the input graph is one of the parameters. These results give the first known bounds for such types of problems parameterized by clique-width.

First, we consider the Dense k-Subgraph problem (also known as the k-Cluster problem). This problem asks whether, given a graph G and positive integers k and q, there exists an induced subgraph of G with k vertices and at least q edges. Clearly, Dense k-Subgraph is NP-hard since it is a generalization of the k-Clique problem. It remains NP-hard, even when restricted to comparability graphs, bipartite graphs and chordal graphs [12], as well as on planar graphs [26]. Polynomial algorithms were given for cographs, split graphs [12], and for graphs of bounded tree-width [26]. Considerable work has been done on approximation algorithms for this problem [3,4,15,19,27].

Next, we consider some degree-constrained subgraph problems. The objective in such problems is to find a subgraph satisfying certain lower or upper bounds on the degree of each vertex. Typically it is necessary to either check the existence of a subgraph satisfying the degree constraints or to minimize (maximize) some parameter (usually the size of the subgraph).

The d-Regular Subgraph problem, which asks whether a given graph contains a d-regular subgraph, has been intensively studied. We mention here only some complexity results. Chvátal et al. [11] proved that this problem is NP-complete for $d = 3$. It was shown that the problem with $d = 3$ remains NP-complete for planar bipartite graphs with maximum degree four, and that when $d \geq 3$, it is NP-complete even for bipartite graphs with maximum degree at most $d+1$. Some further results were given in [7,32,33,34]. We consider a variant of this problem called d-Regular Induced Subgraph, where we ask whether a given graph G contains a d-regular *induced* subgraph. This variant of the problem has also been studied. In particular, the parameterized complexity of different variants of the problem was considered by Moser and Thilikos [31] and by Mathieson and Szeider [29]. Observe that, trivially, d-Regular Induced Subgraph can be solved in polynomial time for $d \leq 2$, and it easily follows from the known hardness results for d-Regular Subgraph that d-Regular Induced Subgraph is NP-complete for any fixed $d \geq 3$.

In [2] Amini et al. introduced the Minimum Subgraph of Minimum Degree at least d problem. This problem asks whether, given a graph G and positive integers d and k, there exists a subgraph of G with at most k vertices and minimum degree at least d. The parameterized complexity of the problem was considered in [2]. Some other hardness and approximation results can be found in [1].

Our Main Results and the Organization of the Paper. In Section 2 we give some basic definitions and some preliminary results. In Section 3 we consider the Dense k-Subgraph and Sparse k-Subgraph problems. The Sparse k-Subgraph problem is dual to Dense k-Subgraph and it asks whether, given a graph G and positive integers k and q, there exists an induced subgraph of G with k vertices and at most q edges. We prove that these problems can be solved in time $k^{O(w)} \cdot n$ for n-vertex graphs of clique-width at most w if an expression tree of width w is given, but they cannot be solved in time $2^{o(w \log k)} \cdot n^{O(1)}$ unless ETH fails even if an expression tree of width w is included in the input. In Section 4 we consider the d-Regular Induced Subgraph and Minimum Subgraph of Minimum Degree at least d problems. We construct dynamic programming algorithms which solve these problems in time $d^{O(w)} \cdot n$ for n-vertex graphs of clique-width at most w if an expression tree of width w is given, and then prove that these problems cannot be solved in time $2^{o(w \log d)} \cdot n^{O(1)}$ unless ETH fails even if an expression tree of width w is provided. We conclude the paper with some open problems.

2 Definitions and Preliminary Results

Graphs. We 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 $v \in V(G)$, $E_G(v)$ denotes the set of edges incident with v. The *degree* of a vertex v is denoted by $d_G(v)$. For a non-negative integer d, a graph G is called *d-regular* if all vertices of G have degree d. For a graph G, the *incidence graph* of G is the bipartite graph $I(G)$ with 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. We denote by $\overline{G}$ the *complement* of a graph G, i.e. the graph with vertex set $V(G)$ such that any two distinct vertices are adjacent in $\overline{G}$ if and only if they are non-adjacent in G. For a set of vertices $S \subseteq V(G)$, $G[S]$ denotes the subgraph of G induced by S, and by $G - S$ we denote the graph obtained from G by the removal of all the vertices of S, i.e. the subgraph of G induced by $V(G) \setminus S$.

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

An *expression 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, and they correspond to initial w-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 w-graph resulting from the relabeling operation $\rho_{i \to j}$ applied to the graph corresponding to the child.
- A join node $\eta_{i,j}$ has one child and is associated with the w-graph resulting from the join operation $\eta_{i,j}$ applied to 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 *width* of the tree T is the number of different labels appearing in T. If a graph G has $\mathbf{cwd}(G) \le w$ then it is possible to construct a rooted expression tree T of G with *width* w. Given a node X of an expression tree, the graph G_X is the graph formed by the subtree of the expression tree rooted at X.

Parameterized Reductions. We refer the reader to the books [18,21] for a detailed treatment of parameterized complexity. Here we only define the notion of parameterized (linear) reduction, which is the main tool for establishing our results. For parameterized problems A, B, 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 which transforms an instance (x, k) of A into an instance $(x', g(k))$ of B in time $f(k)|x|^\alpha$ so that $(x, k) \in A$ if and only if $(x', g(k)) \in B$. The reduction is called *linear* if $g(k) = O(k)$.

Capacitated Domination. For our reductions we use a variant of the Capacitated Dominating Set problem. The parameterized complexity of this problem, with the tree-width of the input graph being the parameter, was considered in [5,16].

A *red-blue capacitated graph* is a pair (G, c), where G is a bipartite graph with a vertex bipartition into sets 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 the set 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* $f\colon B \to S$ which maps every vertex in B 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 $v \in S$, vertices in the set $f^{-1}(v)$ are *dominated by* v. The Red-Blue Capacitated Dominating Set (or Red-Blue CDS) problem asks whether, given a red-blue capacitated graph (G, c) and a positive integer k, 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 next proposition immediately follows from the results proved in [22].

Proposition 1. *The* Red-Blue CDS *and* Red-Blue Exact Saturated CDS *problems cannot be solved in time $f(w) \cdot n^{o(w)}$, where n is the number of vertices of the input graph G and w is the clique-width of the incidence graph $I(G)$, unless ETH fails, even if an expression tree of width w for $I(G)$ is given.*

The proof of Proposition 1 uses the result of Chen et al. [8,9,10] that there is no algorithm for k-Clique (finding a clique of size k) running in time $f(k) \cdot n^{o(k)}$ unless there exists an algorithm for solving 3-SAT running in time $2^{o(n)}$ on a formula with n variables. Proposition 1 was proved via a linear reduction from the k-Multi-Colored Clique problem (see [5,22]). The k-Multi-Colored Clique 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 in G. It should be noted that the construction of an expression tree of bounded width is part of the reduction and it is done in polynomial time. Lokshtanov, Marx and Saurabh [28] considered a special restricted variant of k-Multi-Colored Clique called $k \times k$-Clique. In this variant of the problem $|V_1| = \ldots = |V_k| = k$. They proved the following.

Proposition 2 ([28]). *The $k \times k$-*Clique *problem cannot be solved in time $2^{o(k \log k)} \cdot n^{O(1)}$, where n is the number of vertices of the input graph G, unless ETH fails.*

By replacing k-Multi-Colored Clique by the $k \times k$-Clique problem in the reductions used for the proof of Proposition 1, we obtain the following corollary.

Corollary 1. *The* Red-Blue CDS *and* Red-Blue Exact Saturated CDS *problems cannot be solved in time $2^{o(w \log n)} \cdot n^{O(1)}$, where n is the number of vertices of the input graph G and w is the clique-width of the incidence graph $I(G)$, unless ETH fails, even if an expression tree of width w for $I(G)$ is given.*

Observe that Corollary 1 gives a slightly stronger claim than Proposition 1: while $o(w) \cdot \log n = o(w \log n)$, it is not so the other way around.

3 Sparse and Dense k-Subgraph Problems

In this section we consider the Dense k-Subgraph and Sparse k-Subgraph problems. The aim of this section is the proof of the following theorem.

Theorem 1. *The* Sparse k-Subgraph *problem can be solved in time $k^{O(w)} \cdot n$ on n-vertex graphs of clique-width at most w if an expression tree of width w is given, but it cannot be solved in time $2^{o(w \log k)} \cdot n^{O(1)}$ unless ETH fails, even if an expression tree of width w is given.*

Clearly, Sparse k-Subgraph and Dense k-Subgraph are dual, i.e. Sparse k-Subgraph is equivalent to Dense k-Subgraph for the complement of the input graph. Since for any graph G, $\mathbf{cwd}(\overline{G}) \leq 2 \cdot \mathbf{cwd}(G)$ (see e.g. [14,35]), we can immediately get the following corollary.

Corollary 2. *The* Dense k-Subgraph *problem can be solved in time $k^{O(w)} \cdot n$ on n-vertex graphs of clique-width at most w if an expression tree of width w is given, but it cannot be solved in time $2^{o(w \log k)} \cdot n^{O(1)}$ unless ETH fails, even if an expression tree of width w is given.*

3.1 Algorithmic Upper Bounds for Sparse k-Subgraph

We sketch a dynamic programming algorithm for solving Sparse k-Subgraph in time $k^{O(w)} \cdot n$ on graphs of clique-width at most w. We describe what we store in the tables corresponding to the nodes in an expression tree.

Let G be a graph with n vertices and let T be an expression tree for G of width w. For a node X of T, let $U_1(X), \ldots, U_w(X)$ be the sets of vertices of G_X labeled $1, \ldots, w$, respectively. The table of data for the node X contains entries which store a positive integer $p \leq q$ and a vector $(s_1, \ldots, s_w)$ of non-negative integers such that $s = s_1 + \ldots + s_w \leq k$ for $i \in \{1, \ldots, w\}$, for which p is the minimum number of edges of an induced subgraph H with s vertices such that for $i \in \{1, \ldots, w\}$, $s_i = |U_i(X) \cap V(H)|$. If X is the root node of T

then G contains an induced subgraph with k vertices and at most q edges if and only if the table for X contains an entry with the parameter $p \leq q$ and vector $(s_1, \dots, s_w)$ such that $s_1 + \dots + s_w = k$.

The details how the tables are created and updated are omitted here because of the space restrictions. Correctness of the algorithm follows from the description of the procedure.

Since for each X, the table for X contains at most $(k+1)^w$ vectors and for each vector only one value of the parameter p is stored, the algorithm runs in time $k^{O(w)} \cdot n$. This proves that SPARSE k-SUBGRAPH can be solved in time $k^{O(w)} \cdot n$ on graphs of clique-width at most w.

3.2 Lower Bounds

To prove our lower bounds we give a reduction from the RED-BLUE CDS problem parameterized by the clique-width of the incidence graph of the input graph.

Construction. Let (G, c, k) be an instance of RED-BLUE CDS with $R = \{u_1, \dots, u_n\}$ being the set of red vertices and $B = \{v_1, \dots, v_r\}$ being the set of blue vertices. Let m be the number of edges of G. We assume without loss of generality that G has no isolated vertices. Hence, $m \geq n, r$.

First, we construct the auxiliary gadget $F(l)$ for a positive integer l.

Auxiliary gadget $F(l)$***:*** Construct an $l+m+1$-partite graph $K_{2,\dots,2}$ and denote by x_{i1}, x_{i2} the vertices of the i-th set of the partition (see Figure 1).

Reduction: Now we describe our reduction.

1. A copy of a gadget $F(k)$ is constructed. Denote this graph by F_R and let $V(F_R) = \{x_{i1}^R, x_{i2}^R | 1 \leq i \leq k+m+1\}$.
2. For each $i \in \{1, \dots, n\}$, a copy of a gadget $F(c(u_i))$ is created. Denote this graph by F_{u_i} and let $V(F_{u_i}) = \{x_{j1}^{u_i}, x_{j2}^{u_i} | 1 \leq j \leq c(u_i) + m + 1\}$.
3. For each $i \in \{1, \dots, r\}$, a copy of a gadget $F(1)$ is created. Denote this graph by F_{v_i} and let $V(F_{v_i}) = \{x_{j1}^{v_i}, x_{j2}^{v_i} | 1 \leq j \leq m+2\}$.
4. For each $e \in E(G)$, the vertex w_e is constructed.
5. For each $i \in \{1, \dots, n\}$, let $\{e_1, \dots, e_{d_i}\} = E(u_i)$ for $d_i = d_G(u_i)$. We consider the vertices $w_{e_1}, \dots, w_{e_{d_i}}$; these vertices are joined by edges to the vertices x_{i1}^R, x_{i2}^R of F_R, and for each $j \in \{1, \dots, d_i\}$, w_{e_j} is joined by edges to the vertices $x_{j1}^{u_i}, x_{j2}^{u_i}$ of F_{u_i}.
6. For each $i \in \{1, \dots, r\}$, let $\{e_1, \dots, e_{d_i}\} = E(v_i)$ for $d_i = d_G(v_i)$. We consider the vertices $w_{e_1}, \dots, w_{e_{d_i}}$ and for each $j \in \{1, \dots, d_i\}$, w_{e_j} is joined by edges to the vertices $x_{j1}^{v_i}, x_{j2}^{v_i}$ of F_{v_i}.
7. Create $2m+1$ vertices $z_1, \dots, z_{2m+1}$ and join them to all vertices w_e for $e \in E(G)$.

Denote the obtained graph by H (see Figure 1).

Due the space restrictions the proof of the following lemmas are omitted.

Lemma 1. *The red-blue graph G has a capacitated dominating set of size at most k if and only if H contains an induced subgraph with* $2(m+1)(n+r+1) + 2m+1+r$ *vertices and at most* $2m(m+1)(n+r+1) + r(2m+1)$ *edges.*

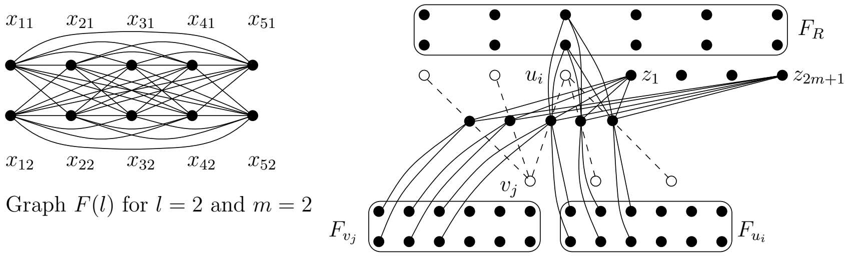

Fig. 1. Construction of H

We prove an upper bound for the clique-width of H as a linear function in the clique-width of the incidence graph $I(G)$ of G.

Lemma 2. *We have* $\mathbf{cwd}(H) \leq 9 \cdot \mathbf{cwd}(I(G)) + 1$ *and an expression tree of width at most* $9 \cdot \mathbf{cwd}(I(G)) + 1$ *for* H *can be constructed in polynomial time given an expression tree of width* $\mathbf{cwd}(I(G))$ *for* $I(G)$*.*

To complete the proof of Theorem 1, notice that the number of vertices of H and the parameter k are polynomial in $n + r$. Therefore, $\log k$ is linear in $\log(n + k)$, and if we could solve Sparse k-Subgraph in time $2^{o(\mathbf{cwd}(H)\log k)} \cdot |V(H)|^{O(1)}$ then Red-Blue CDS could be solved in time $2^{o(\mathbf{cwd}(I(G))\log|V(G)|)} \cdot |V(G)|^{O(1)}$. By Corollary 1, it cannot be done unless ETH fails.

4 Degree-Constrained Subgraph Problems

The first aim of this section is the proof of the following theorem.

Theorem 2. *The* d-Regular Induced Subgraph *problem can be solved on* n*-vertex graphs of clique-width at most* w *in time* $d^{O(w)} \cdot n$ *if an expression tree of width* w *is given for the input graph, but it cannot be solved in time* $2^{o(w \log d)} \cdot n^{O(1)}$ *unless ETH fails, even if an expression tree of width* w *is given.*

Proof. The algorithmic upper bounds are proved by constructing a dynamic programming algorithm for solving d-Regular Induced Subgraph in time $d^{O(w)} \cdot n$ on graphs of clique-width at most w. To prove our complexity lower bound, we give a reduction from the Red-Blue Exact Saturated CDS problem, parameterized by the clique-width of the incidence graph of the input graph, to the d-Regular Induced Subgraph problem. The proof is organized as follows: we first give a construction, then prove its correctness and finally bound the clique-width of the transformed instance.

Construction. Let (G, c, k) be an instance of Red-Blue Exact Saturated 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. Let $d = n+r+1$ if $n+r$ is even and let $d = n+r+2$ otherwise; notice that d is odd. We need an auxiliary gadget.

Auxiliary gadget $F(x)$**:** Let x be a vertex. We construct $\frac{d-1}{2}$ copies of K_{d+1}, subdivide one edge of each copy, and glue (identify) all these vertices of degree two into one vertex y. Finally we join x and y by an edge. We are going to attach gadgets $F(x)$ to other parts of our construction through the vertex x. This vertex is called the *root* of $F(x)$. The gadget $F(x)$ for $d = 5$ is illustrated in Figure 2.

Reduction: Now we describe our reduction. Let $s = d - r - 1$ and $t = d - k - 1$.

1. Vertices $u_1, \ldots, u_n$ are created.
2. A clique of size r with vertices $v_1, \ldots, v_r$ is constructed.
3. For each edge $e = u_i v_j$ of G, a vertex w_e is added, joined by edges to u_i and v_j, and $d-2$ copies of $F(w_e)$ are constructed.
4. A clique of size s with vertices $a_1, \ldots, a_s$ is created, all vertices a_i are joined to vertices $v_1, \ldots, v_r$, and for each $i \in \{1, \ldots, s\}$, a copy of $F(a_i)$ is added.
5. A vertex x is introduced and joined by edges to $v_1, \ldots, v_r$ and $a_1, \ldots, a_s$.
6. A vertex y is added and joined by an edge to x, and $k-1$ copies of $F(y)$ are added.
7. A clique of size t with vertices $b_1, \ldots, b_t$ is constructed, the vertex y is joined by edges to all vertices of the clique, and for each $j \in \{1, \ldots, t\}$, k copies of $F(b_i)$ are added.
8. A vertex z is introduced and joined by edges to vertices y and $b_1, \ldots, b_t$.
9. For each $i \in \{1, \ldots, n\}$, we let $l_i = d - c(u_i) - 1$ and do the following:
   - Add a vertex p_i, join it to z by an edge, and construct $c(u_i) - 1$ copies of $F(p_i)$.
   - Construct a clique of size l_i with vertices $c_{i1}, \ldots, c_{il_i}$, join them to the vertex p_i by edges, and for each $j \in \{1, \ldots, l_i\}$, introduce $c(u_i)$ copies of $F(c_{ij})$.
   - Join the vertex u_i to the vertices p_i and $c_{i1}, \ldots, c_{il_i}$ by edges.

Denote the obtained graph by H. The construction of H is illustrated in Figure 2.

The proof of the following lemmas are omitted.

Lemma 3. *The red-blue graph G has an exact saturated capacitated dominating set of size k if and only if H contains an induced d-regular subgraph.*

Now we show that the clique-width of H is bounded from above by a linear function in the clique-width of the incidence graph $I(G)$ of G.

Lemma 4. *We have that* $\mathbf{cwd}(H) \leq 3 \cdot \mathbf{cwd}(I(G)) + 6$ *and an expression tree of width at most* $3 \cdot \mathbf{cwd}(I(G)) + 6$ *for H can be constructed in polynomial time assuming we are given an expression tree of width* $\mathbf{cwd}(I(G))$ *for* $I(G)$.

To conclude this part of the proof of Theorem 2, we observe that the number of vertices of H and the parameter d are polynomial in $n + r$, and therefore if we could solve d-Regular Induced Subgraph in time $2^{o(\mathbf{cwd}(H) \log d)} \cdot |V(H)|^{O(1)}$ then the Red-Blue Exact Saturated CDS could be solved in time $2^{o(\mathbf{cwd}(I(G)) \log |V(G)|)} \cdot |V(G)|^{O(1)}$. By Corollary 1, this cannot be done unless ETH fails. $\square$

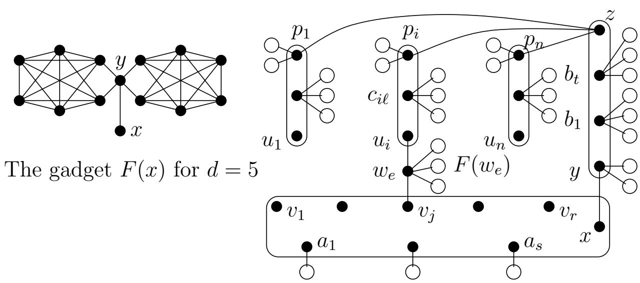

Fig. 2. Construction of H

In the d-Regular Induced Subgraph problem we ask about the existence of a d-regular induced subgraph for a given graph. It is possible to get similar results for some variants of this problem. The Minimum d-Regular Induced Subgraph problem and the Maximum d-Regular Induced Subgraph problem are respectively the problems of finding a d-regular induced subgraph of minimum and maximum size. For the Counting d-Regular Induced Subgraph problem, we are interested in the number of induced d-regular subgraphs of the input graph. Using Theorem 2 we get the following corollary.

Corollary 3. *The* Minimum d-Regular Induced Subgraph, Maximum d-Regular Induced Subgraph *and* Counting d-Regular Induced Subgraph *problems can be solved on n-vertex graphs of clique-width at most w in time $d^{O(w)} \cdot n$ if an expression tree of width w is given, but they cannot be solved in time $2^{o(w \log n)} \cdot n^{O(1)}$ unless ETH fails, even if an expression tree of width w is given.*

We conclude this section by considering the Minimum Subgraph of Minimum Degree at least d problem.

Theorem 3. *The* Minimum Subgraph of Minimum Degree at least d *problem can be solved on n-vertex graphs of clique-width at most w in time $d^{O(w)} \cdot n$ if an expression tree of width w is given, but it cannot be solved in time $2^{o(w \log d)} \cdot n^{O(1)}$ unless ETH fails, even if an expression tree of width w is given.*

5 Conclusion

We established tight algorithmic lower and upper bounds for some double-parameterized subgraph problems when the clique-width of the input graph is one of the parameters. We believe that similar bounds could be given for other problems. Another interesting task is to consider problems parameterized by other width-parameters. Throughout the paper, in all our results we assumed that an expression tree of the given width is part of the input. This is crucial, since — unlike the case of tree-width — to date we are unaware of an efficient

(FPT or polynomial) algorithm for computing an expression tree with a constant factor approximation of the clique-width. The algorithm given by Oum and Seymour in [30] provides a constant factor approximation for another graph parameter — *rank-width* [24,30]. Hence, it is natural to ask whether it is possible to establish tight algorithmic bounds for DENSE k-SUBGRAPH, d-REGULAR INDUCED SUBGRAPH and MINIMUM SUBGRAPH OF MINIMUM DEGREE AT LEAST d parameterized by the rank-width of the input graph. Also it would be interesting to consider problems parameterized by the tree-width. For example, it can be shown that d-REGULAR INDUCED SUBGRAPH and MINIMUM SUBGRAPH OF MINIMUM DEGREE AT LEAST d can be solved in time $d^{O(t)} \cdot n$ for n-vertex graphs of tree-width at most t. Is this bound asymptotically tight?

References

1. Amini, O., Peleg, D., Pérennes, S., Sau, I., Saurabh, S.: Degree-Constrained Subgraph Problems: Hardness and Approximation Results. In: Bampis, E., Skutella, M. (eds.) WAOA 2008. LNCS, vol. 5426, pp. 29–42. Springer, Heidelberg (2009)
2. Amini, O., Sau, I., Saurabh, S.: Parameterized Complexity of the Smallest Degree-Constrained Subgraph Problem. In: Grohe, M., Niedermeier, R. (eds.) IWPEC 2008. LNCS, vol. 5018, pp. 13–29. Springer, Heidelberg (2008)
3. Arora, S., Karger, D.R., Karpinski, M.: Polynomial time approximation schemes for dense instances of p-hard problems. In: STOC, pp. 284–293. ACM (1995)
4. Asahiro, Y., Iwama, K., Tamaki, H., Tokuyama, T.: Greedily Finding a Dense Subgraph. In: Karlsson, R., Lingas, A. (eds.) SWAT 1996. LNCS, vol. 1097, pp. 136–148. Springer, Heidelberg (1996)
5. Bodlaender, H.L., Lokshtanov, D., Penninkx, E.: Planar Capacitated Dominating Set is $W[1]$-Hard. In: Chen, J., Fomin, F.V. (eds.) IWPEC 2009. LNCS, vol. 5917, pp. 50–60. Springer, Heidelberg (2009)
6. Cai, L., Juedes, D.W.: On the existence of subexponential parameterized algorithms. J. Comput. Syst. Sci. 67, 789–807 (2003)
7. Cheah, F., Corneil, D.G.: The complexity of regular subgraph recognition. Discrete Applied Mathematics 27, 59–68 (1990)
8. Chen, J., Chor, B., Fellows, M., Huang, X., Juedes, D., Kanj, I.A., Xia, G.: Tight lower bounds for certain parameterized NP-hard problems. Information and Computation 201, 216–231 (2005)
9. Chen, J., Huang, X., Kanj, I.A., Xia, G.: On the computational hardness based on linear FPT-reductions. J. Comb. Optim. 11, 231–247 (2006)
10. Chen, J., Huang, X., Kanj, I.A., Xia, G.: Strong computational lower bounds via parameterized complexity. J. Comput. Syst. Sci. 72, 1346–1367 (2006)
11. Chvátal, V., Fleischner, H., Sheehan, J., Thomassen, C.: Three-regular subgraphs of four-regular graphs. J. Graph Theory 3, 371–386 (1979)
12. Corneil, D.G., Perl, Y.: Clustering and domination in perfect graphs. Discrete Appl. Math. 9, 27–39 (1984)
13. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory Comput. Syst. 33, 125–150 (2000)
14. Courcelle, B., Olariu, S.: Upper bounds to the clique width of graphs. Discrete Appl. Math. 101, 77–114 (2000)

15. Demaine, E.D., Hajiaghayi, M.T., Kawarabayashi, K.: Algorithmic graph minor theory: Decomposition, approximation, and coloring. In: FOCS, pp. 637–646. IEEE Computer Society (2005)
16. Dom, M., Lokshtanov, D., Saurabh, S., Villanger, Y.: Capacitated Domination and Covering: A Parameterized Perspective. In: Grohe, M., Niedermeier, R. (eds.) IWPEC 2008. LNCS, vol. 5018, pp. 78–90. Springer, Heidelberg (2008)
17. Downey, R.G., Estivill-Castro, V., Fellows, M.R., Prieto, E., Rosamond, F.A.: Cutting up is hard to do: the parameterized complexity of k-cut and related problems. Electr. Notes Theor. Comput. Sci. 78 (2003)
18. Downey, R.G., Fellows, M.R.: Parameterized complexity. Monographs in Computer Science. Springer, New York (1999)
19. Feige, U., Peleg, D., Kortsarz, G.: The dense k-subgraph problem. Algorithmica 29, 410–421 (2001)
20. Fellows, M.R., Rosamond, F.A., Rotics, U., Szeider, S.: Clique-Width is NP-Complete. SIAM J. Discrete Math. 23, 909–939 (2009)
21. Flum, J., Grohe, M.: Parameterized complexity theory. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin (2006)
22. Fomin, F.V., Golovach, P.A., Lokshtanov, D., Saurabh, S.: Algorithmic lower bounds for problems parameterized by clique-width. In: SODA, pp. 493–502 (2010)
23. Hlinený, P., Oum, S.-I.: Finding branch-decompositions and rank-decompositions. SIAM J. Comput. 38, 1012–1032 (2008)
24. Hlinený, P., Oum, S.-I., Seese, D., Gottlob, G.: Width parameters beyond tree-width and their applications. Comput. J. 51, 326–362 (2008)
25. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? J. Comput. Syst. Sci. 63, 512–530 (2001)
26. Keil, J.M., Brecht, T.B.: The complexity of clustering in planar graphs. J. Combin. Math. Combin. Comput. 9, 155–159 (1991)
27. Kortsarz, G., Peleg, D.: On choosing a dense subgraph (extended abstract). In: FOCS, pp. 692–701. IEEE (1993)
28. Lokshtanov, D., Marx, D., Saurabh, S.: Slightly superexponential parameterized problems. In: SODA (2011)
29. Mathieson, L., Szeider, S.: The parameterized complexity of regular subgraph problems and generalizations. In: CATS. CRPIT, vol. 77, pp. 79–86. Australian Computer Society (2008)
30. Oum, S.-I., Seymour, P.: Approximating clique-width and branch-width. J. Combin. Theory Ser. B 96, 514–528 (2006)
31. Moser, H., Thilikos, D.M.: Parameterized complexity of finding regular induced subgraphs. J. Discrete Algorithms 7, 181–190 (2009)
32. Stewart, I.A.: Deciding whether a planar graph has a cubic subgraph is np-complete. Discrete Mathematics 126, 349–357 (1994)
33. Stewart, I.A.: Finding regular subgraphs in both arbitrary and planar graphs. Discrete Applied Mathematics 68, 223–235 (1996)
34. Stewart, I.A.: On locating cubic subgraphs in bounded-degree connected bipartite graphs. Discrete Mathematics 163, 319–324 (1997)
35. Wanke, E.: k-NLC graphs and polynomial algorithms. Discrete Appl. Math. 54, 251–266 (1994)