Minimum Conflict Free Colouring Parameterized by Treewidth

Pradeesha Ashok[1(✉)], Rathin Bhargava[1], Naman Gupta[2], Mohammad Khalid[1], and Dolly Yadav[1]

[1] International Institute of Information Technology Bangalore, Bangalore, India
pradeesha@iiitb.ac.in,
{rathin.bhargava,mohammadkhalid.udayagiri,dolly.yadav}@iiitb.org
[2] Indian Institute of Science Education and Research Mohali, Mohali, India
ms17169@iisermohali.ac.in

Abstract. Conflict free q-Colouring of a graph G refers to the colouring of a subset of vertices of G using q colours such that every vertex has a neighbour of unique colour. In this paper, we study the Minimum Conflict free q-Colouring problem. Given a graph G and a fixed constant q, Minimum Conflict free q-Colouring is to find a Conflict free q-Colouring of G that minimises the number of coloured vertices. We study the Minimum Conflict free q-Colouring problem parameterized by the treewidth of G. We give an FPT algorithm for this problem and also prove running time lower bounds under Exponential Time Hypothesis (ETH) and Strong Exponential Time Hypothesis (SETH).

Keywords: Conflict free colouring of graphs · Parameterized complexity · FPT algorithms · Treewidth · Exponential Time Hypothesis · Strong Exponential Time Hypothesis

1 Introduction

Given a graph $G(V, E)$ a q-colouring refers to a function $c : V \rightarrow [q]$, where $[q] = \{1, 2, \ldots, q\}$. A well studied colouring problem in graphs is the *Proper Colouring* problem which is a colouring c with the added constraint that if $(u, v) \in E$ then $c(u) \neq c(v)$. Many other versions of graph colouring are also studied. In this paper, we study the Conflict Free Colouring problem in graphs.

Given a hypergraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, a Conflict free q-colouring of $\mathcal{G}$ refers to a colouring $c : \mathcal{V} \rightarrow [q]$ such that every hyperedge $E \in \mathcal{E}$ has a vertex v with a distinct colour $c(v)$ i.e., no other vertex in E has the colour $c(v)$ under c. Conflict free colouring was initially studied for geometric hypergraphs motivated by the frequency allocation problem in wireless networks [6]. Later, Pach and Tardos [10] studied this problem for hypergraphs induced by graph neighbourhoods. In this version, all vertices of the graph are coloured. Abel et al. [1] studied a closely

M. Changat and S. Das (Eds.): CALDAM 2020, LNCS 12016, pp. 439–450, 2020.
https://doi.org/10.1007/978-3-030-39219-2_35

related problem of colouring only a subset of vertices of G such that for every vertex in V there exists a vertex with a distinct colour in its neighbourhood. They studied algorithmic and combinatorial problems on Conflict free colouring of planar and outerplanar graphs. [1] also studied the bicriteria problem of minimizing the number of coloured vertices in a Conflict free q-colouring of graphs. We study this problem for general graphs.

We now state the problem that we study. Consider a graph $G(V, E)$ and a fixed constant q. Let $N(v)$ denote the open neighbourhood of a vertex v i.e., the set of all vertices u in V such that $(u, v) \in E$ and $N[v]$ denote the closed neighbourhood of v i.e., $N[v] = N(v) \cup \{v\}$. A Closed Neighbourhood Conflict Free q-Colouring is a colouring c of a subset V' of V such that for every vertex $v \in V$, there exists a vertex $u \in N[v]$ such that $c(u) \neq c(u')$ for any vertex $u' \in N[v] \setminus \{u\}$. Similarly, a Open Neighbourhood Conflict Free q-Colouring is a colouring c of a subset V' of V such that for every vertex $v \in V$, there exists a vertex $u \in N(v)$ such that $c(u) \neq c(u')$ for any vertex $u' \in N(v) \setminus \{u\}$. We study the following minimisation problems.

Min-q-CNCF: Given a graph $G(V, E)$ and a fixed constant q, find a Closed Neighbourhood Conflict Free q-Colouring that minimises the number of coloured vertices.

Min-q-ONCF: Given a graph $G(V, E)$ and a fixed constant q, find a Open Neighbourhood Conflict Free q-Colouring that minimises the number of coloured vertices.

The above problems can be seen as variants of an important problem in graph theory called the Minimum Dominating Set problem. Specifically, when $q = 1$, Min-q-CNCF and Min-q-ONCF respectively are the Efficient Dominating Set problem and Perfect Dominating Set problem. Therefore Min-q-CNCF and Min-q-ONCF are NP-hard [7,12].

We study the parameterized complexity of the minimum Conflict Free q-colouring problem when parameterized by the treewidth τ of the graph and prove upper and lower bounds.

1. We show that Min-q-CNCF and Min-q-ONCF are FPT when parameterized by treewidth. This can also be proved using Courcelle's theorem [3]. We give a constructive proof by giving an algorithm with running time $\mathcal{O}(q^{O(\tau)})$ for both problems.
2. For $q = 1$, we show that an algorithm with running time $\mathcal{O}(2^{o(|V|)})$ cannot exist for Min-q-CNCFand Min-q-ONCF , under Exponential Time Hypothesis. Since $|V|$ is an upper bound for τ, this also rules out the possibility of algorithms with running time $\mathcal{O}(2^{o(\tau)})$. For $q = 2$, we show that an algorithm with running time $\mathcal{O}(2^{o(|V|)})$ cannot exist for Min-q-CNCF and we show that an algorithm with running time $\mathcal{O}(2^{o(\tau)})$ cannot exist for Min-q-ONCF.
3. For $q \geq 3$, we show that an algorithm with running time $\mathcal{O}((q-\epsilon)^{o(\tau)})$ cannot exist for Min-q-CNCF and Min-q-ONCF , under Strong Exponential Time Hypothesis.

2 Preliminaries

In this section, we give definitions and results that will be used in subsequent sections.

Parameterized Complexity: Parameterized complexity was introduced as a technique to design efficient algorithms for problems that are NP-hard. An instance of a parameterized problem is a pair (Π, k) where Π is the input and k is the parameter. A parameter is a positive integer that represents the value of a fixed attribute of the input or output and is assumed to be much smaller than the size of the input, n. A parameterized problem is said to be *fixed parameter tractable* (FPT) if there exists an algorithm that solves it in $f(k)n^{O(1)}$ time, where f is a computable function independent of n. Refer [4,5] for a detailed description of Parameterized Complexity. We denote an FPT running time using the notation $\mathcal{O}(f(k))$ that hides the polynomial functions.

Exponential Time Hypothesis (ETH) [4]**:** For $q \geq 3$, let δ_q be the infinimum of the set of constants of c for which there exists an algorithm solving the n variable q-SAT in time $\mathcal{O}(2^{cn})$. ETH states that $\delta_3 > 0$. **Strong Exponential Time Hypothesis (SETH)** states that $\lim_{q\to\infty} \delta_q = 1$. In other words, ETH implies that 3-SAT cannot be solved faster than $\mathcal{O}(2^{o(n)})$ and SETH implies q-SAT cannot be solved faster than $\mathcal{O}((q-\epsilon)^n)$.

Treewidth [4]**:** A tree decomposition is a pair $\mathcal{T} = (T, \{X_t\}_{t\in V(T)})$ where T is a tree whose every node t is assigned to a vertex subset, $X_t \subseteq V(G)$, called a bag, such that the following conditions hold.

- $\bigcup_{t\in V(T)} X_t = V(G)$. In other words, every vertex of G is at least in one bag.
- For every $(u, v) \in E(G)$, there exists a node t of T such that bag X_t contains both u and v.
- For every node $u \in V(G)$, the set $T_u = \{t \in V(T) : u \in X_t\}$, i.e. the set of nodes whose corresponding bags contain u, induces a connected subtree of T. The width of the tree decomposition $\mathcal{T}$ is the maximum size of the bag minus 1. The *treewidth* of the graph G, denoted by $\tau(G)$ is the minimum possible width of a tree decomposition of G.

Nice Tree Decomposition: A rooted tree decomposition, $(T, \{X_t\}_{t\in V(T)})$ is nice if

- $X_r = \emptyset$ and $X_l = \emptyset$ where r is the root and l is a leaf of the tree.
- Every other node of T is one of the following
- **Introduce node**: A node t with exactly one child t' such that $X_t = X_{t'} \cup \{v\}$ where $v \notin X_t$
- **Forget node**: A node t with exactly one child t' such that $X_t = X_{t'} \backslash \{w\}$
- **Join node**: A node t with two children t_1, t_2 such that $X_t = X_{t_1} = X_{t_2}$.
- **Introduce edge node**: A node t that introduces the edge (u, v) where $u, v \in X_t$ and has only one child t' such that $X_t = X_{t'}$.

In this variant of the tree decomposition, the total number of the nodes is still $O(\tau n)$. It is known that we can compute a nice tree decomposition $(T, \mathcal{X})$ of G with $|V(T)| \in |V(G)|^{O(1)}$ of width at most 5τ in time $O(2^{O(\tau)}n)$, where τ is the treewidth of G [4].

Positive 1-in-3 SAT Problem: Given a 3-CNF formula ϕ with all positive literals, the Positive 1-in-3 SAT problem asks whether there exists a truth assignment such that exactly one literal is true in all clauses.

For a given graph G, let $\chi_{CF}(G)$ represent the minimum value of q such that there exists a Conflict free q-colouring of G. In a conflict free colouring c of $V' \subseteq V$, if a vertex v has a neighbour $u \in V'$ such that $c(u)$ is unique in the neighbourhood of v, then v is said to be *conflict-free dominated* by u.

3 FPT Algorithm for Min-q-CNCF

We present an FPT algorithm for the Min-q-CNCF problem parameterized by treewidth. Our algorithm uses a popular FPT technique known as Dynamic Programming over Treewidth. Assume a nice tree composition $\mathcal{T}$ of G is given. For a node t in $\mathcal{T}$, let X_t represent the set of vertices in the bag of t. With each node t of the tree decomposition we associate a subgraph G_t of G defined as: $G_t = (V_t, E_t = \{e : e \text{ is introduced in the subtree rooted at t}\})$. Here, V_t is the union of all bags present in the subtree rooted at t.

For every node t, we define colouring functions α, β, f where $\alpha : X_t \rightarrow \{c_0, c_1, ..., c_q\}$, $\beta : X_t \rightarrow \{c_1, ..., c_q\}$ and $f : X_t \rightarrow \{B, W, C, R\}$. Here, c_i represents the i^{th} colour for $1 \leq i \leq q$ and c_0 denotes a no-colour assignment. $\alpha(u)$ and $\beta(u)$ denotes the colour of the vertex u and the colour it is dominated by respectively. The function f denotes the 'state' of each vertex. We now give a little more insight to what α, β and f represent. For any vertex $u \in X_t$ we have the following.

A *Black* vertex is denoted by $f(u) = B$. Intuitively, a black vertex is coloured and dominated in G_t. A *Cream* vertex is denoted by $f(u) = C$. A cream vertex is coloured but not dominated in G_t. A *White* vertex is denoted by $f(u) = W$ and is not coloured but is dominated in G_t. A *Grey* vertex is denoted by $f(u) = R$. It is not coloured and not dominated in G_t. A tuple $[t, \alpha, \beta, f]$ is *valid* if the following conditions are true for every vertex $u \in X_t$.

- If $f(u) = B$ then $\alpha(u) \neq c_0$.
- If $f(u) = C$ then $\alpha(u) \neq c_0$ and $\alpha(u) \neq \beta(u)$.
- If $f(u) \in \{W, R\}$ then $\alpha(u) = 0$.

A colouring $c : V_t \rightarrow \{c_0, c_1, ..., c_q\}$ is said to extend $[t, \alpha, \beta, f]$ if every vertex in $V_t \setminus X_t$ is conflict-free dominated and for every $v \in X_t$, the following is true:

1. $c(v) = \alpha(v)$.
2. if $f(v) \in \{B, W\}$, then v has exactly one neighbour u in G_t such that $c(u) = \beta(v)$.
3. if $f(v) \in \{C, R\}$ then no neighbour of v in G_t is given the colour $\beta(v)$ by c.

We now define sub problems for every node t. Let $dp[t, \alpha, \beta, f]$ denote the minimum number of coloured vertices in any colouring of V_t that extends $[t, \alpha, \beta, f]$. Every tuple, which is either invalid or cannot be extended to a conflict free colouring, corresponds to $dp[t, \alpha, \beta, f] = \infty$.

We define $f_{v\to\gamma}$ where $\gamma \in \{B, W, R, C\}$, as the function where $f_{v\to\gamma}(x) = f(x)$, if $x \neq v$, and $f_{v\to\gamma}(x) = \gamma$, otherwise. Similarly, we define $\alpha_{v\to\gamma}$, for $\gamma \in \{c_0, c_1, \ldots, c_q\}$ and $\beta_{v\to\gamma}$ for $\gamma \in \{c_1, c_2, \ldots, c_q\}$. We now give recursive formulae for $dp[., ., ., .]$.

Leaf Node: In this case $X_t = \phi$. So, $dp[t, \phi, \phi, \phi] = 0$.

Introduce Vertex Node: Let t' be the only child node of t. Then, $\exists\ v \notin X_{t'}$ such that $X_t = X_{t'} \cup \{v\}$.

$$dp[t, \alpha, \beta, f] = \begin{cases} dp[t', \alpha|_{X_{t'}}, \beta|_{X_{t'}}, f|_{X_{t'}}] + 1 & \text{if } f(v) = B \wedge \alpha(v) = \beta(v). \\ dp[t', \alpha|_{X_{t'}}, \beta|_{X_{t'}}, f|_{X_{t'}}] + 1 & \text{if } f(v) = C \wedge \alpha(v) \neq \beta(v). \\ dp[t', \alpha|_{X_{t'}}, \beta|_{X_{t'}}, f|_{X_{t'}}] & \text{if } f(v) = R. \\ \infty & \text{otherwise.} \end{cases}$$

The correctness of the recurrence formula follows from the fact that a vertex is an isolated vertex when it is introduced and can be conflict-free dominated only by itself.

Forget Vertex Node: Let t' be the only child node of t. Then, $\exists v \notin X_t$ such that $X_{t'} = X_t \cup \{v\}$. The vertex v cannot be dominated by a vertex introduced above X_t. Hence $[t, \alpha, \beta, f]$ cannot be extended by a colouring if $f(v) \in \{C, R\}$. Hence we get the following:

$$dp[t, \alpha, \beta, f] = \min_{1\leq i,j\leq q} \begin{cases} dp[t', \alpha_{v\to c_i}, \beta_{v\to c_j}, f_{v\to B}]. \\ dp[t', \alpha_{v\to c_0}, \beta_{v\to c_i}, f_{v\to W}]. \end{cases}$$

Introduce Edge Node: Let t be an introduce edge node with child node t'. Let (u^*, v^*) be the edge introduced at t. Consider distinct $u, v \in \{u^*, v^*\}$. We decide the value of $dp[t, \alpha, \beta, f]$ based on the following cases.

$$dp[t, \alpha, \beta, f] = \begin{cases} dp[t', \alpha, \beta, f_{u\to C, v\to C}] & ((f(u), f(v)) = (B, B) \wedge (\alpha(u) = \beta(v)) \wedge (\alpha(v) = \beta(u))). \\ dp[t', \alpha, \beta, f_{v\to C}] & (f(u) \in \{B, C\} \wedge f(v) = B \wedge \alpha(u) = \beta(v) \wedge \alpha(v) \neq \beta(u)). \\ dp[t', \alpha, \beta, f_{v\to R}] & (f(u) \in \{B, C\} \wedge f(v) = W \wedge \alpha(u) = \beta(v)). \\ \infty & (f(u) \in \{B, C\} \wedge f(v) \in \{C, R\} \wedge \alpha(u) = \beta(v)). \\ dp[t', \alpha, \beta, f] & otherwise. \end{cases}$$

Clearly, the edge (u, v) can only dominate v if u is coloured with $\beta(v)$. If v is conflict-free dominated by u and $f(v) \in \{W, B\}$, then v was not conflict-free dominated in t' under the same colouring functions α and β. Hence if $f(v)$ is black (or white), we set $f(v)$ to cream (or grey) in the child node.

Join Node: Let t be a join node with 2 child nodes t_1, t_2 and $X_t = X_{t_1} = X_{t_2}$. We call tuples $[t_1, \alpha_1, \beta_1, f_1]$ and $[t_2, \alpha_2, \beta_2, f_2]$ as $[t, \alpha, \beta, f]$-consistent if the following conditions hold for all $v \in X_t$.

- $\alpha(v) = \alpha_1(v) = \alpha_2(v)$.
- $\beta(v) = \beta_1(v) = \beta_2(v)$.
- If $f(v) = B$ then $(f_1(v), f_2(v)) = (B, B) \wedge \alpha(v) = \beta(v)$ or $(f_1(v), f_2(v)) \in \{(B, C), (C, B)\} \wedge \alpha(v) \neq \beta(v))$.
- If $f(v) = C$ then $f_1(v) = f_2(v) = C$.
- If $f(v) = R$ then $f_1(v) = f_2(v) = R$.
- If $f(v) = W$ then $(f_1(v), f_2(v)) \in \{(W, G), (G, W)\}$.

All other colourings are not consistent. For example, assume $f_1(v) = f_2(v) = W$ and both $dp[t_1, \alpha_1, \beta_1, f_1]$ and $dp[t_2, \alpha_2, \beta_2, f_2]$ are finite. Then v is conflict free dominated in G_{t_1} and G_{t_2}. By the property of nice tree decomposition, an edge between two vertices in a join node is introduced above the join node. Hence X_t induces an independent set in G_t. Therefore v is conflict free dominated by a vertex outside X_t in both G_{t_1} and G_{t_2}. Now, in G_t, v has two neighbours with colour $\beta(v)$ and hence v cannot be conflict free dominated.

Now we give the recurrence formula for $dp[]$.

$$dp[t, \alpha, \beta, f] = \min \left(dp[t_1, \alpha_1, \beta_1, f_1] + dp[t_2, \alpha_2, \beta_2, f_2] - |f^{-1}(B)| - |f^{-1}(C)| \right)$$

where tuples $[t_1, \alpha_1, \beta_1, f_1]$ and $[t_2, \alpha_2, \beta_2, f_2]$ are $[t, \alpha, \beta, f]$-consistent.

Now $dp[r, \emptyset, \emptyset, \emptyset]$ where r is the root of $\mathcal{T}$ gives the desired solution. Also, it can be seen that all recurrences except those for join nodes, can be computed in $\mathcal{O}((4q^2)^\tau)$ time. For a join node t, two tuples are consistent with $[t, \alpha, \beta, f]$ if (f, f_1, f_2) is in one of 7 forms. Thus, processing a join node can be done in $\mathcal{O}((7q^2)^\tau)$ time. Hence, we get the following result.

Theorem 1. *There exists an FPT algorithm with running time $\mathcal{O}(q^{O(\tau)})$ for* Min-q-CNCF *parameterized by the treewidth of the graph.*

We also prove a similar result for Min-q-ONCF . The algorithm is very similar to that given above and can be found in the full version of the paper.

Theorem 2. *There exists an FPT algorithm with running time $\mathcal{O}(q^{O(\tau)})$ for* Min-q-ONCF *parameterized by the treewidth of the graph.*

4 Lower Bounds

In this section, we give lower bounds that complement the results given in Sect. 3.

4.1 Lower Bounds for Min-q-CNCF

Theorem 3. Min-1-CNCF *cannot be solved in $\mathcal{O}(2^{o(n)})$ time, unless ETH fails.*

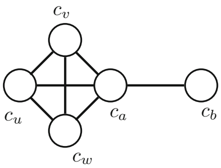


Fig. 1. Clause gadget

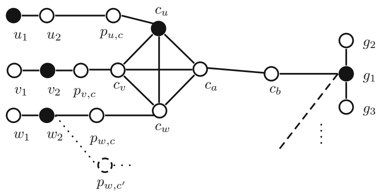


Fig. 2. Combination of vertex and clause gadgets

We prove the theorem by giving a linear reduction from the Positive 1-in-3 SAT problem. It is known that Positive 1-in-3 SAT cannot be solved in $\mathcal{O}(2^{o(n)})$ time, unless ETH fails [9,11]. Let ϕ be an instance of the Positive 1-in-3 SAT problem, with n variables and m clauses. We will construct a graph $G(V, E)$ corresponding to ϕ. For every variable u of ϕ, we add two nodes u_1, u_2 to $V(G)$ and the edge (u_1, u_2) to $E(G)$. For every clause c, we add a gadget as shown in Fig. 1.

If a variable u belongs to a clause c, then in G, the vertex u_2 is connected to one of the vertices in $\{c_u, c_v, c_w\}$ in the clause gadget of c, through a connector vertex $p_{u,c}$ as shown in Fig. 2. The vertex c_b in each clause gadget is connected to a *global vertex* g_1. The global vertex also has two neighbours of degree 1, g_2 and g_3. Clearly G has $2n + 8m + 3$ vertices.

Lemma 1. *ϕ is satisfiable if and only if G can be conflict-free coloured using one colour.*

Proof. Assume that ϕ has a satisfying assignment. We now give a valid conflict free 1-colouring of G. Colour the global vertex g. If a variable u is true in the satisfying assignment, then we colour the vertex u_1 of the corresponding variable gadget, otherwise we colour the vertex u_2. Observe that if the vertex u_1 is coloured, then u_2 is conflict-free dominated by u_1 and hence, u_2 cannot be coloured. For the same reason, a connector vertex $p_{u,c}$ that connects u_2 to the clause gadget of clause c cannot be coloured. Therefore, in order to conflict-free dominate the vertex $p_{u,c}$, the vertex c_u of the clause gadget should be coloured. By similar arguments, if the vertex u_1 in variable gadget is uncoloured then the corresponding vertex c_u in clause gadget should also be uncoloured.

Let c be an arbitrary clause in ϕ and let u, v, w be the variables in c. In a satisfying assignment, exactly one among u, v, w is true. Without loss of generality, let u be the variable that is true. Then c_u is coloured and is conflict free dominated by itself. c_v, c_w and c_a are uncoloured but conflict free coloured by c_u. This means that for every clause c in ϕ the vertex c_b of the corresponding clause gadget in G is uncoloured and is conflict-free dominated by g. This gives us a valid 1-conflict free colouring of graph G.

Similarly we can prove that if G has a valid 1-conflict free colouring then ϕ has a satisfying assignment. □

Now we will consider the case $q = 2$. Assume the colours used are red and blue.

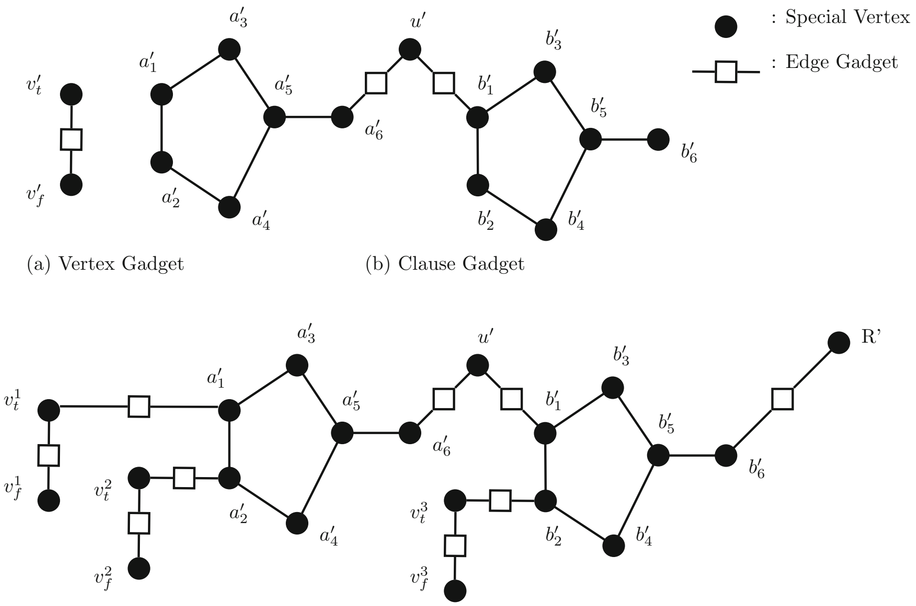
Theorem 4. MIN-2-CNCF *cannot be solved in* $\mathcal{O}(2^{o(n)})$ *time, unless ETH fails.*

We use the following lemma from [1].

Lemma 2. *[Lemma 3.2, [1]] Let* G *be any graph,* $u,\ v \in V(G)$ *and* $e = (v, u) \in E(G)$*. If* $N(v)$ *contains two disjoint and independent copies of a graph* $H = G_q$ *with* $\chi_{CF}(H) = q$*, not adjacent to any other vertex* $w \in G$*, every* q*-conflict-free colouring of* G *colours* v*. If the same holds for* u *and in addition,* $N_G(u) \cap N_G(v)$ *contains two disjoint and independent copies of a graph* $J = G_{q-1}$ *with* $\chi_{CF}(J) = q - 1$*, not adjacent to any other vertex* $w \in G$*, every* q*-conflict-free colouring of* G *colours* u *and* v *with different colours.*

We're looking at the special case, where $q = 2$. As given in Lemma 2, G_1 is a single vertex. G_2 is $K_{1,3}$ with one edge subdivided by another vertex.

We define a vertex $v' \in V(G)$ as a *special vertex* if $N(v')$ contains two disjoint and independent copies of G_2. We also define an edge gadget between 2 special vertices u' and v' as a *special edge* between u' and v' if $(u', v') \in E(G)$ and $N_G(u) \cap N_G(v)$ contains two disjoint and independent copies of G_1. We note that by Lemma 2, a special vertex in a graph needs to be coloured *red* or *blue* and if there exists a special edge between two special vertices u' and v', then u' and v' needs to be coloured with opposite colours.



(a) Vertex Gadget (b) Clause Gadget

(c) Vertex Gadget v_i connected to Clause Gadget C_j connected to the palette vertex R'

Fig. 3. All the gadgets used in this proof

We give a linear reduction from the 3-Sat problem to Min-2-CNCF . Let ϕ be an instance of the 3-Sat problem with n variables and m clauses. We construct an instance of Min-2-CNCF , $G(V, E)$, corresponding to ϕ. For every variable v in ϕ, we add a vertex gadget to G which consists of two special vertices v'_t and v'_f connected by a special edge. We define a sub-clause gadget which consists of 6 special vertices denoted as $a'_1, a'_2, a'_3, a'_4, a'_5, a'_6$ where $a'_i,\ i \in [5]$ form C_5 - $\{a'_1, a'_3, a'_5, a'_4, a'_2, a'_1\}$. The vertex a'_6 is adjacent to a'_5. For every clause c in ϕ, we add a clause gadget to G. A clause gadget is constructed by taking 2 sub-clause gadgets $\{a'_1, \ldots, a'_6\}$ and $\{b'_1, \ldots, b'_6\}$ and connecting them through a vertex u' by adding special edges between a'_6 and u', and b'_1 and u'. Let the variables in the clause c_i be denoted as $v^i_k, k \in [3]$. Then the variable gadgets corresponding to the the variables v^i_1, v^i_2, v^i_3 in G are respectively connected to the vertices a'_1, a'_2, b'_2 of the clause gadget corresponding to c_i, through special edges. If the variable v appears in c_i as a positive literal, then v_f is connected to the clause gadget, otherwise v_t is connected. Finally, there exists a palette vertex R such that there is a special edge between R and b_6 for all clause gadgets. (Refer Fig. 3). Since every vertex gadget and clause gadget has a constant number of vertices, V has $O(n + m)$ vertices. Specifically, V contains $k = 2n + 13m + 1$ special vertices.

Lemma 3. *Let vertices a'_1 and a'_2 have exactly 1 neighbour outside the sub-clause gadget which is coloured the opposite to it and conflict-free dominates itself. Then, given that we colour only special vertices, if a'_1 and a'_2 are both coloured red, a'_6 will be coloured red. If a'_1 or a'_2 is coloured blue, then there exists a colouring where a'_6 will be coloured blue.*

Proof. We first prove that colouring a'_1 and a'_2 *red* forces a'_6 to be coloured *red*. By construction of the sub-clause gadget, a'_1 and a'_2 are adjacent to each other. It's given that both a'_1 and a'_2 have one blue neighbour outside the gadget. The colour *red* appears twice in the closed neighbourhood of a'_1. Thus, the only colour which can dominate a'_1 is blue. If a'_3 is coloured blue, a'_1 would not be conflict free dominated. Thus, a'_3 is coloured *red*. By a similar argument involving a'_2, we can show that a'_4 has to be coloured *red*. In this way, let a'_5 be coloured *blue*. By contradiction, if a'_5 is coloured *red*, then a'_3 does not have a conflict free neighbour as all its neighbours are *red*. Thus, a'_6 gets coloured red. If it gets coloured blue, then a'_5 will have 2 red and 2 blue neighbours in its closed neighbourhood and cannot get dominated. Thus, a'_6 is coloured *red*. If both a'_1 and a'_2 are coloured blue, then we can prove, in a similar case to the one done above, that a'_6 will be coloured blue.

To prove the second part, let's assume without loss of generality, that a'_1 is coloured *red* and a'_2 is coloured *blue*. Since a'_1 is connected to a blue neighbour outside the sub-clause gadget, a'_3 is forced to be coloured *blue*. Likewise, a'_4 is coloured *red*. To ensure that a'_6 gets a *blue* colour, we need a'_5 to be coloured *blue*. It can be seen that this is a valid colouring. □

Lemma 4. *Any instance ϕ of* 3-Sat *is satisfiable if and only if G can be conflict free 2-coloured with at most k coloured vertices.*

Proof. Assume there exists a 2-conflict free colouring of G which colours at most k vertices. We will show that ϕ has a satisfying assignment. By Lemma 2, in any conflict free 2-colouring of G every special vertex is coloured. Since we have coloured at most k vertices, only the special vertices are coloured. The palette vertex, R, is coloured since it is a special vertex. Without loss of generality, let it be coloured red. Since b'_6 vertices of all the clause gadgets have special edges to R, they are coloured *blue*. By Lemma 3, we know that at least one of a'_1, a'_2, b'_2 is coloured blue. Let that one vertex be u. Now, u is connected to a corresponding variable gadget. If it is connected to v'_t, then assign that variable v in ϕ false, otherwise assign true. It is easy to see that this is a satisfying assignment. Similarly, we can see that if there exists a satisfying assignment of ϕ, there exists a conflict free 2-colouring of G. □

Now the theorem follows from ETH. □

Now we consider $q \geq 3$.

Lemma 5. *For $q \geq 3$, an algorithm with running time $\mathcal{O}((q-\epsilon)^{o(\tau)})$ cannot exist for* Min-q-CNCF *, under Strong Exponential Time Hypothesis.*

We reduce Proper q-colouring to Min-q-CNCF. We know from [8] that Proper q-colouring under SETH, cannot be solved faster than $\mathcal{O}((q-\epsilon)^{\tau(G)})$. As shown in Lemma 3.4 from [1], from any graph G, we can construct G' which can be Conflict free q-coloured if and only if G can be proper q-coloured. From Claim 2, Lemma 7 from [2], we know that the treewidth of G' is $max\{\tau(G), q\}$. where $\tau(G)$ is the treewidth of the graph G. Hence, Min-q-CNCF colouring cannot be solved faster than $\mathcal{O}((q-\epsilon)^{max\{\tau(G),q\}})$ and the lemma follows. □

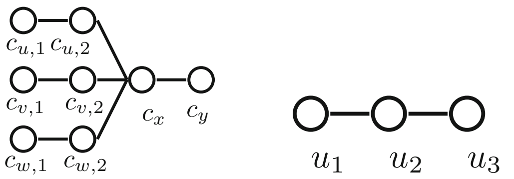
4.2 Lower Bounds for Min-q-ONCF

Theorem 5. Min-1-ONCF *cannot be solved in $\mathcal{O}(2^{o(n)})$ time, unless ETH fails.*

Proof. We give a reduction from Positive 1-in-3 SAT. Let ϕ' be an instance of Positive 1-in-3 SAT with n variables and m clauses. We will construct a graph $G'(V, E)$ corresponding to ϕ'. For each variable and clause we construct a variable gadget and a clause gadget respectively. The variable gadget for an arbitrary variable u is a path of length 3. The clause gadget is shown in Fig. 4. There is a global gadget which consists of a path of length 4, $g_1 - g_2 - g_3 - g_4$ with a pendant vertex connected at vertex g_3. If a variable u belongs to a clause c, then vertex u_3 is connected to vertex $c_{u,1}$ in G'. Vertex g_4 is connected to vertex c_y of all clause gadgets. Clearly, G' has $8m + 3n + 5$ vertices.

Lemma 6. *ϕ' is satisfiable if and only if G' can be conflict free* 1*-coloured.*

Proof. We will show that if G' has a valid 1-conflict free colouring then ϕ' has a satisfying truth assignment. We can show the other direction by similar arguments. Observe that the vertices $g_2,\ g_3$ should always be coloured and $g_1,\ g_5,\ g_4$



(a) Clause gadget for clause c

(b) variable gadget for variable u

Fig. 4. Clause and variable gadgets

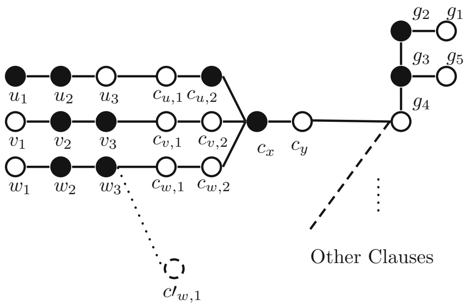


Fig. 5. Combined gadget

should always be uncoloured as it is the only way to conflict free colour the vertices g_1, g_2, g_3, g_4 and g_5. Since g_4 is uncoloured and is dominated by g_3, all its neighbours except g_3 should also be uncoloured and dominated in a valid colouring. Hence, for every clause c, the vertex c_y in G' should be uncoloured and c_x should be coloured. To conflict free dominate c_x, exactly one among the vertices $c_{u,2}$, $c_{v,2}$, $c_{w,2}$ should be coloured. Without loss of generality, assume that the vertex $c_{u,2}$ is coloured. Since $c_{u,2}$ is coloured, the vertex u_1 should be coloured and the vertices v_1, w_1 should be uncoloured. Then, assign true to variable u and false for variables v and w. It can be seen that this gives a satisfying assignment for ϕ' (Fig. 5). □

Theorem 6. MIN-2-ONCF *cannot be solved in* $\mathcal{O}(2^{o(\tau)})$*, assuming ETH is true.*

Proof. [2] gives a result that a variant of MIN-2-ONCF where every vertex needs to be coloured cannot be solved in time $\mathcal{O}(2^{o(\tau)})$ under ETH. For proving this result, they give a reduction from the 3-SAT problem that reduces an instance of the 3-SAT problem, ϕ, to an instance of the MIN-2-ONCF problem, G, with treewidth a linear function of $|\phi|$. We use the same reduction and modify G by connecting a vertex of degree 1 to every vertex in G. Note that the treewidth of the modified instance, G', is still a linear function of $|\phi|$. Now, we will show that ϕ has a satisfying assignment if and only if G' has a valid conflict free 2-colouring that colours at most $|V(G)|$ vertices. This follows from the result in [2] and the fact that a vertex of degree one can only be conflict free dominated by its neighbour when open neighbourhood is considered.

Lemma 7. MIN-Q-ONCF *cannot be solved in time* $\mathcal{O}((q-\epsilon)^{\tau(G)})$ *under SETH.*

Proof. We give a reduction from the PROPER Q-COLOURING problem to the MIN-Q-ONCF problem. We know from [8] that PROPER Q-COLOURING under SETH, cannot be solved faster than $\mathcal{O}((q-\epsilon)^{\tau(G)})$. We consider the graph G' in lemma 5 in [2] and construct graph G'' by adding a vertex with degree 1 to all the vertices in G'. Now the proof follows as before.

References

1. Abel, Z., et al.: Conflict-free coloring of graphs. SIAM J. Discrete Math. **32**(4), 2675–2702 (2018)
2. Bodlaender, H.L., Kolay, S., Pieterse, A.: Parameterized complexity of conflict-free graph coloring. In: Algorithms and Data Structures - 16th International Symposium, WADS 2019, Edmonton, AB, Canada, 5–7 August 2019, Proceedings, pp. 168–180 (2019). https://doi.org/10.1007/978-3-030-24766-9_13
3. Courcelle, B.: The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. Inf. Comput. **85**(1), 12–75 (1990)
4. Cygan, M., et al.: Parameterized Algorithms, vol. 4. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21275-3
5. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, Switzerland (2012). https://doi.org/10.1007/978-3-319-21275-3
6. Even, G., Lotker, Z., Ron, D., Smorodinsky, S.: Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks. SIAM J. Comput. **33**(1), 94–136 (2003)
7. Kratochvíl, J., Křivánek, M.: On the computational complexity of codes in graphs. In: Chytil, M.P., Koubek, V., Janiga, L. (eds.) MFCS 1988. LNCS, vol. 324, pp. 396–404. Springer, Heidelberg (1988). https://doi.org/10.1007/BFb0017162
8. Lokshtanov, D., Marx, D., Saurabh, S.: Known algorithms on graphs of bounded treewidth are probably optimal. In: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 777–789. SIAM (2011)
9. Muzi, I., O'Brien, M.P., Reidl, F., Sullivan, B.D.: Being even slightly shallow makes life hard. In: 42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017), vol. 83, p. 79. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2017)
10. Pach, J., Tardos, G.: Conflict-free colourings of graphs and hypergraphs. Comb. Probab. Comput. **18**(5), 819–834 (2009)
11. Schaefer, T.J.: The complexity of satisfiability problems. In: Proceedings of the Tenth Annual ACM Symposium on Theory of Computing, pp. 216–226. ACM (1978)
12. Yen, C.C., Lee, R.C.T.: The weighted perfect domination problem. Inf. Process. Lett. **35**(6), 295–299 (1990)