Freie Universität Berlin

Bachelor Thesis at the Institute for Computer Science at Freie Universität Berlin

Work group theoretical computer science

Understanding and implementation of the algorithm for three-coloring in triangle-free planar graphs

Qianli Wang

Matriculation number: 5338778

qiaw99@zedat.fu-berlin.de

Advisor:	Prof. Dr. László Kozma
First Examiner:	Prof. Dr. László Kozma
Second Examiner:	PD Dr. rer. nat. Klaus Kriegel

Berlin, April 24, 2021

Abstract

Grötzsch's theorem affirms that every planar graph G in which there is no triangle is 3-colorable. Dvorak, Kawarabayashi, and Thomas have designed a linear-time algorithm relying on the Grötzsch's theorem to find a proper 3-coloring of the given graph. Compared to Kowalik who used a data structure called "Short Path Data Structure (SPDS)", it avoids complex data structures, which makes it easier to implement. However, because of the difficulty of their paper and complicated proofs, it's really hard to understand it. So we will explain their main idea and proofs with more illustrations and in detail.

Affidavit

I hereby affirm that this work was written by anyone else other than myself. All used resources such as reports, books, websites or similar are listed in the bibliography. Quotations from external works are marked as well. The thesis has not been submitted or published to any other examination commission in the same or a similar form.

April 24, 2021

Qianli Wang

Contents

1 Introduction

The graph coloring problem is one of the most famous *NP-complete* problems. Given a graph $G = (V, E)$ and k, where V is the set of vertices in the graph, E is the set of edges and k is the number of available colors. The problem is to assign a certain color to every vertex $v \in V$ with the constraint: there is no pair of vertices which are adjacent and have the same color. This problem has many applications:

(1) *Map Coloring:* Coloring geographical maps of countries or states where no two adjacent countries can be assigned same color. Four colors are enough to color.

(2) *Sudoku:* Sudoku is a variation of graph coloring problem where every cell represents a vertex. There is an edge between two vertices if they are in same row or same column or same block.

(3) *Register Allocation:* In compiler optimization, register allocation is the process of assigning a large number of target program variables onto a small number of CPU registers. [8]

Definition 1.1. A *graph* is a tuple $G = (V, E)$ consisting of a nonempty set V of vertices and a set of edges E. [1]

Definition 1.2. A *vertex coloring* of a graph $G = (V, E)$ is a map $c : V \longrightarrow S$ such that $c(v) \neq c(w)$ whenever v and w are adjacent. The elements of the set S are called the available colors. If G has a k-COLORING, namely the size of S is k, then we call the graph G is k-colorable. [2]

Claim 1. *2-COLORING problem can be solved in polynomial time.*

Proof. Suppose the given graph $G = (V, E)$ and color c_1, c_2.

(1) Randomly pick $v \in V$, color it with c_1.

(2) Apply BFS starting with vertex v and color its neighbors with c_2. The point is that for $\forall u \in V$, we should color its neighbors with the other color alternatively: assume that the color of u is c_1, then we assign c_2 to its neighbors, and vice versa.

(3) After finishing BFS, go through all vertices again to check whether there is a vertex that is assigned with same colors other than its neighbor(s). If *yes*, then the graph is *not* 2-colorable, *otherwise* it is 2-colorable.

The running time is similar to BFS: $\mathcal{O}(|V| + |E|)$. □

Observation 1. *We can check whether a graph is bipartite by coloring the graph using two colors. If a given graph is 2-colorable, then it is bipartite.*

Observation 2. *k-COLORING problems are* $NP - complete$ *for* $k \geq 3$.

As previously stated, the k-COLORING problem is NP-complete, which means that it seems hardly possible to have a polynomial time algorithm for this problem. Moving on now to consider 3-COLORING problem under circumstance that there is no triangle within the given graph G. Grötzsch's theorem [4] states that each triangle-free planar graph is 3-colorable. Thomassen [9] has also found two proofs and extended the result in various way, by which a quadratic algorithm for finding suitable 3-coloring can be developed possibly. Kowalik [6] maintains a complex data structure called *Short Path Data Structure (SPDS)*, which will be built in linear time and enables that finding shortest paths of length at most 2 in planar graph takes $\mathcal{O}(1)$ time. The SPDS will be constantly updated during the particular sequence of operations. And its running time for finding 3-COLORING is $\mathcal{O}(n \log n)$. After that, Dvorak, Kawaravayashi and Thomas [3] have designed a linear-time algorithm which still relies on the Grötzsch's theorem but avoids complex data structures. Nevertheless, their paper is quite complicated for readers. So this thesis will give a deeper and detailed view into their paper for better understanding of their main ideas and proofs.

In the second section, we will give some needed definitions and give an overview of this linear-time algorithm. In the third section, we will introduce a attribute called safety with which it's possible to reduce the size of graph. Then in the fourth section, we will give a short proof of Grötzsch's theorem by two lemmas. Next, we will give an overview how a naive algorithm for three-coloring is implemented and show some improvements. In the last two sections, the way to have a linear-time algorithm will be given and its correctness will be also proven.

2 Fundamentals

2.1 Basic definitions in graph theory

Before we start with the proofs of algorithm, let's give some useful definitions in graph theory:

Definition 2.1. A *cycle* C in G is a simple path $v_1, v_2, ..., v_n$, where $v_n = v_1$. In the **Figure 1**, $v_1v_2v_3v_4$ is a cycle.

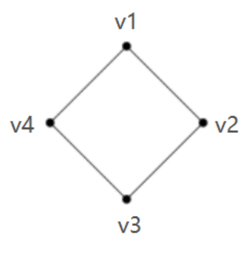

Figure 1: Cycle C

Definition 2.2. When a connected graph can be drawn without any edges crossing, it is called *planar*. When a planar graph is drawn in this way, it divides the plane into regions called *faces*. [7] In **Figure 2**, $v_2v_3v_4v_5$ bounds a face.

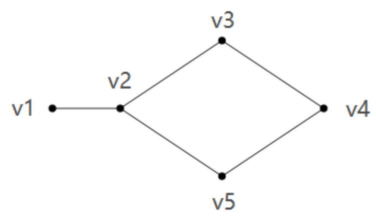

Figure 2: Face F

Definition 2.3. A cycle F is a *facial cycle* in G if it bounds a face in a component in G(possibly the outer face), regardless of whether F itself is a face or not. [3] Both cycles in **Figure 3** are facial cycle.

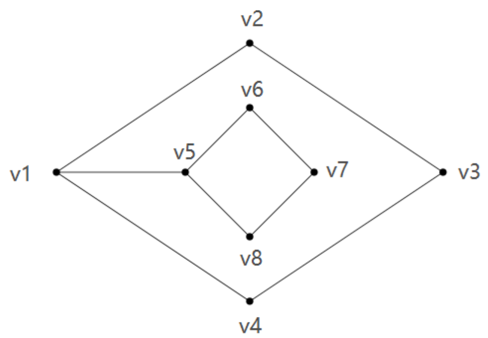

Figure 3: Both cycles $v_1v_2v_3v_4$ and $v_5v_6v_7v_8$ are facial

Definition 2.4. A cycle C in a connected graph G is a *separating cycle* if the deletion of C from G results in a disconnected graph. [10] In the **Figure 4**, $v_1v_2v_3v_4$ is a separating cycle, because inside this cycle there are two more vertices v_6, v_7.

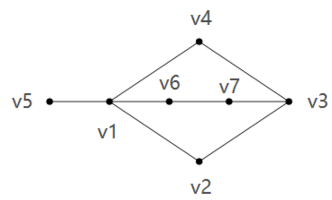

Figure 4: Separating cycle

Observation 3. *In a drawing of a graph, a cycle is either a facial face or a separating cycle.*

Definition 2.5. A vertex v is *incident* with an edge e if $v \in e$; then e is an edge at v. The two vertices incident with an edge are its endvertices or ends, and an edge joins its ends. [2]

Definition 2.6. In the plane, a face f is *the outer face*, if all vertices are contained inside in this face.

Definition 2.7. A *open disk* is a cycle and everything inside it in a drawing of graph but not vertices on the bound. A *closed disk* is a cycle and everything inside it in a drawing of graph including vertices on the bound.

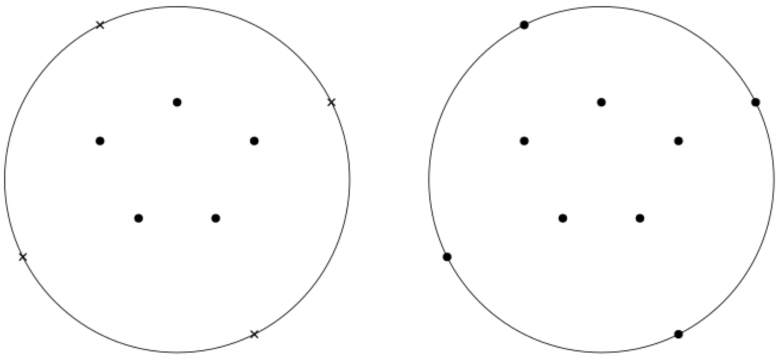

Figure 5: Open and closed disks

Definition 2.8. We say that $P = v_1v_2v_3...v_k$ is an *induced path* in G if $\forall v_iv_j \in E \iff |i-j| = 1$.

2.2 The core idea

After knowing these beneficial definitions, we will introduce five reducible configurations, also called *multigrams*. With such multigrams, we can reduce the size of graph G to have a smaller graph $G^{'}$ by identifying some vertices (see definition in the next section). Meanwhile, although we get a smaller graph $G^{'}$, its coloring can be reconstructed to the coloring of G in constant time. The algorithm detects recursively such multigrams with certain constraints so that the size of graph will always get smaller each time until it's so simple to find the corresponding coloring. Next, according to the resulted coloring, we can reconstruct the coloring of its previous graph step by step. In the end, we'll get the proper coloring of the input graph. [1]

3 Safety of multigram

Let's introduce some reducible multigrams: A *tetragram* is a sequence (v_1, v_2, v_3, v_4) of vertices of G such that it can build a facial cycle in this listed order. Analogously, we can define a *hexagram* (v_1, v_2, v_3, v_4, v_5, v_6). And a *pentagram* (v_1, v_2, v_3, v_4, v_5) is also defined likewise but with the limitation: v_1, v_2, v_3, v_4 have degree exactly three.

3.1 Safe multigrams

As was pointed out in the second section of this thesis, the detected multigrams should possess some attributes, which is called *safety*. Assume that k = 4, 5, 6 and

[1]Demonstration of the algorithm can be found in appendix

$(v_1, v_2, ..., v_k)$ be a tetra-, penta- or hexagram in a triangle-free planar graph G. On the occasion that $k = 4$ or $k = 6$, the tetragram or hexagram is *safe* if every path in G of length at most three with ends v_1 and v_3 is a subgraph of cycle $v_1v_2...v_k$, which means the path(s) from v_1 to v_3 of length at most three has to be part of multigrams. The safety of pentagram is bit complicated: let x_i be the neighbor of v_i different from v_{i-1} and v_{i+1}, where $v_0 = v_5$. Thus $x_i \notin \{v_1, v_2, v_3, v_4, v_5\}$ for the reason that if that is the case, the vertices v_i, x_i form a triangle with v_{i-1} or v_{i+1}, which is a contradiction to the triangle-free graph. A pentagram $(v_1, v_2, v_3, v_4, v_5)$ is *safe* [3], if

- the vertices x_1, x_2, x_3, x_4 are pairwise distinct and pairwise non-adjacent, and
- there is no path in $G \backslash \{v_1, v_2, v_3, v_4\}$ of length at most three from x_2 to v_5, and
- every path in $G \backslash \{v_1, v_2, v_3, v_4\}$ of length at most three from x_3 to x_4 has length exactly two, and its completion via the path $x_3v_3v_4x_4$ results in a facial cycle of length five in G.

Definition 3.1. Given a graph $G = (V, E)$, *identifying a pair of vertices* means that the two selected vertices $u, v \in V(G)$ will be "glued" as a vertex s and the neighbor of s is the union of u, v's neighbors. After that we'll then obtain a new graph $G^{'} = (V^{'}, E^{'})$, where $V^{'} = V \backslash \{u, v\} \cup \{s\}$ and $E^{'}$ will be obtained by deleting parallel edges from E after gluing the vertices. It results $|V(G^{'})| = |V(G)| - 1$.

3.2 Identifying vertices and reconstruction of graph

If there exists a safe tetra- or hexagram, $G^{'}$ can be obtained by identifying vertices v_1 with v_3. v_1 in **Figure 6, 7** is at the same time v_3 as well. For *tetragram* (v_1, v_2, v_3, v_4):

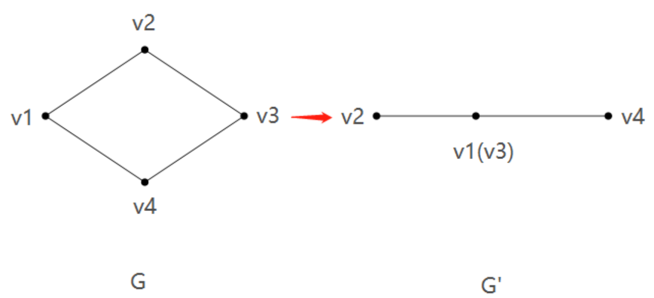

Figure 6: Identifying v_1 and v_3 in a safe tetragram

Reconstruction step: In graph $G^{'}$, we can assign colors as follows: let c_1 be the color of v_2, c_2 be the color of v_1 and c_3 be the color of v_4. Clearly that we can also color v_3 with c_2, since v_1 and v_3 are not adjacent and the tetragram is safe.

For *hexagram* $(v_1, v_2, v_3, v_4, v_5, v_6)$:

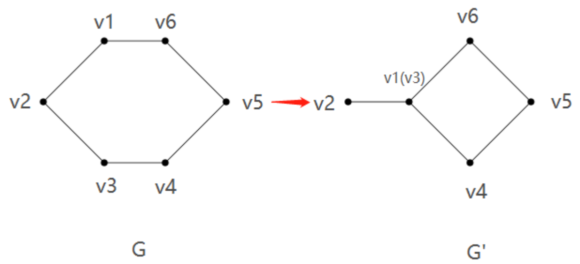

Figure 7: Identifying v_1 and v_3 in a safe hexagram

Reconstruction step: In graph $G^{'}$, we can assign colors following these principles: let c_1 be the color of v_2, c_2 be the color of v_1. We can determine colors of v_4 and v_6 arbitrarily. Assume that the color of v_4 and v_6 is c_3 and the color of v_5 is c_1. Apparently, the color c_2 can be designated for v_3 as well, since v_1 and v_3 are not adjacent and the hexagram is safe.

The case of pentagram (v_1, v_2, v_3, v_4, v_5) is significantly different. $G^{'}$ will be attained from $G \backslash \{v_1, v_2, v_3, v_4\}$ by identifying v_5 with x_2 and x_3 with x_4.

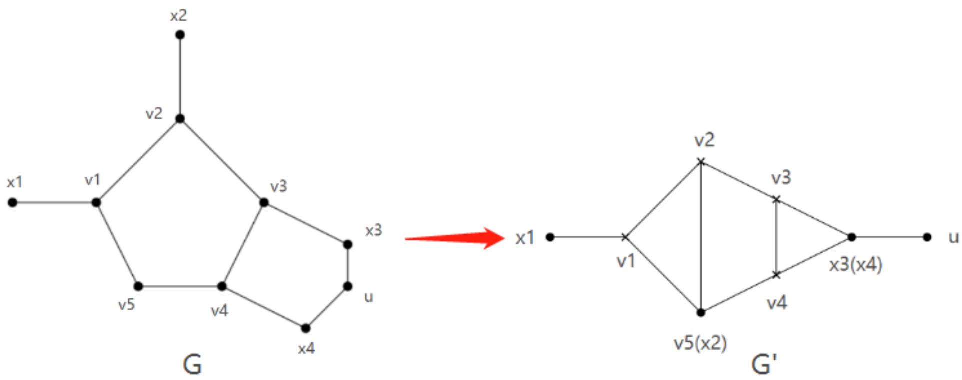

Figure 8: Identifying v_5, x_2 and x_3, x_4 in a safe pentagram

Reconstruction step: In graph $G^{'}$, let c_1 be the color of x_1, c_2 the color of x_2 and v_5, and c_3 the color of x_3 and x_4. Consider the following cases:

- $c_1 = c_2$, we will color the remaining vertices in this order (v_4, v_3, v_2, v_1). Since there are at most two remaining colors when v_i is colored, we can simply choose the third color for it.

- $c_2 = c_3$, similarly, we will color the vertices in the reversed order as in the first case.

- $c_1 \neq c_2$ and $c_2 \neq c_3$, we can color v_2 with c_1, v_1 with c_3, v_3 with c_2 and v_4 with c_1.

3.3 Counterexamples: unsafe multigrams

Subsequently, let's take care of the importance of multigrams. What happens, when we identify certain pair of vertices if the given multigrams are not safe? If a tetragram or hexagram is not safe, then there exists a path of length at most three that is not part of tetragram or hexagram.

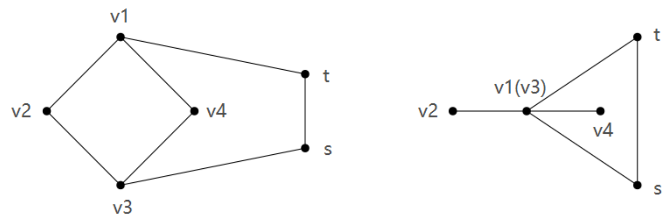

Figure 9: An unsafe tetragram

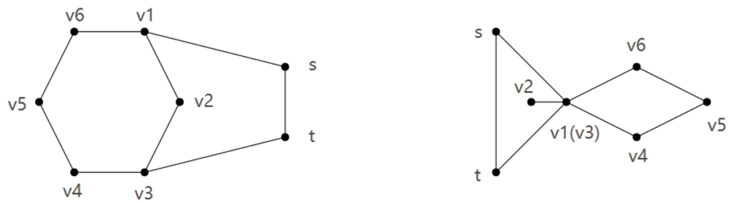

Figure 10: An unsafe hexagram

Observe from above figures that after identifying vertices v_1 with v_3, a triangle consisting of s, t, v_1 will be created.

If a pentagram is not safe, then it has either a path in $G\backslash\{v_1, v_2, v_3, v_4\}$ at most three from x_2 to v_5 or a path in $G\backslash\{v_1, v_2, v_3, v_4\}$ from x_3 to x_4 of length 3.

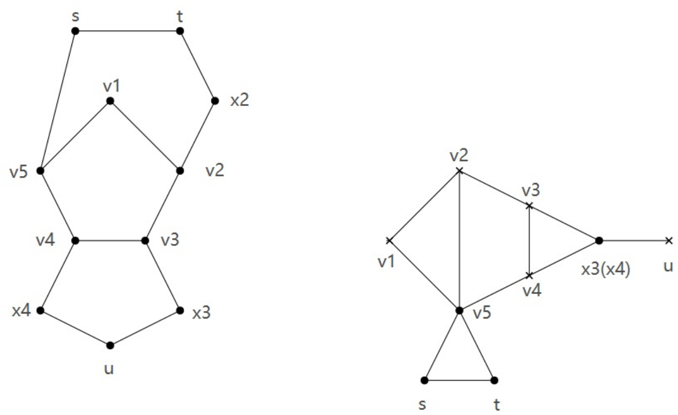

Figure 11: Unsafe pentagram, case 1

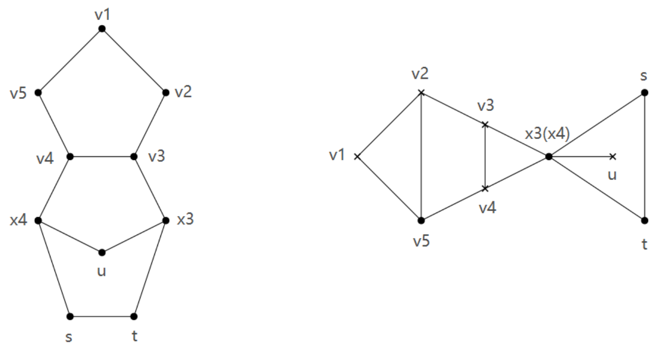

Figure 12: Unsafe pentagram, case 2

Notice that in the former case, $v_5(x_2)$, s and t build a triangle and in the latter case, $x_3(x_4)$, s and t build a triangle in the same way.

4 Proof of Grötzsch's theorem

Firstly, we introduce a method that is often used in graph theory. *Discharging method* is an important proof technique in structural graph theory. The general process of discharging is this: we discharge the graph based on certain "discharging rules," during which some elements gain charges, and some elements lose charges, while the sum of the charges stays constant. [11]

Observation 4. *Given F is a set of all faces embedded in the graph G = (V, E), then:*

$$\sum_{f \in F} deg(f) = 2|E| \ and \ \sum_{v \in V} deg(v) = 2|E|$$

Theorem 4.1 (Euler's formula)**.** For any planar graph with v vertices, e edges, and f faces, we have $v - e + f = 2$. This equation is called *Euler's formula* for planar graphs. [7]

Proof. By induction on the number of edges e.
Base case: Show that the statement holds for the smallest natural number e = 0.
e = 0, v = 1, f = 1 $\rightarrow$ 1 - 0 + 1 = 2.

Inductive step: Show that for any e $\geq$ 0, if $e = n$ holds, then $e = n + 1$ holds as well.
Case 1: G is a tree. It follows $v = e + 1 = n + 2$, $f = 1$. $\longrightarrow v - e + f = (n+2) - (n+1) + 1 = 2$.
Case 2: There exists a simple cycle. Assume that there is a spanning tree T. (u, v) is an edge which does not belong to T. The path between u and v builds a simple cycle with the edge uv. Removal of the edge uv constructs a connect graph $G^{'}$ with $v^{'} = v$, $e^{'} = e - 1$ and $f^{'} = f - 1$.
It leads from induction hypothesis that 2 = $v^{'}$ - $e^{'}$ + $f^{'}$ = $v + (e-1) + (f-1) = v - e + f$. □

Lemma 4.2. Let G be a connected triangle-free plane graph and let f_0 be the unbounded face of G. Assume that the boundary of f_0 is a cycle C of length at most six, and that every vertex of G not on C has degree at least three. If $G \neq C$, then G has either a tetragram, or a pentagram $(v_1, v_2, v_3, v_4, v_5)$ such that $v_1, v_2, v_3, v_4 \notin V(C)$. [3]

Proof. Assign the charge of a vertex v to be 3deg(v) - 12, the charge of the face f_0 to be $3|V(C)|$ + 11 and the charge of a face $f \neq f_0$ of length ℓ to be $3\ell - 12$.

Claim 2. *The sum of the charges of all vertices and faces is -1.*

Proof.

$$
\begin{aligned}
Sum\ of\ charges\ &= \sum_{v \in V}(3deg(v) - 12) + \sum_{f \in F \setminus f_0}(3\ell - 12) + 3|V(C)| + 11 \\
&= \underbrace{3 \cdot 2|E| - 12|V| + 3 \cdot 2|E| - 12|F|}_{\text{apply Euler's formula here: v + f - e = -2}} - (3\ell_{f_0} - 12) + 3|V(C)| + 11 \\
&= -2 \cdot 12 + 3|V(C)| - 3\ell_{f_0} + 12 + 11 \\
&= -1
\end{aligned}
$$

□

Afterwards, we redistribute the charges conforming to the following rules:

(1) Every vertex $v \notin C$ and deg(v) = 3 will receive one unit of charge from each incident face.

(2) Every vertex $v \in C$ and deg(v) = 3 will receive three units from f_0.

(3) Every vertex $v \in C$ and deg(v) = 2 will receive five units from f_0 and one unit from the other incident face.

Claim 3. *The final charge of f_0 is non-negative.*

Proof. Let ℓ be the size of cycle C. As we defined above, the initial charge of face f_0 is $3\ell + 11$. According to **Lemma 4.2**($G \neq C$), we know that there is at least one vertex on C whose degree is at least three, which leads that there are at most $(\ell - 1)$ vertices of C with degree two and one vertex on C has degree at least three. Thus, f_0 can send at most obeying rules (2) and (3) $\big(5(\ell - 1) + 3\big)$ units of charge.
$\longrightarrow$ The final charge of f_0 = $3\ell + 11 - 5(\ell - 1) - 3 = 13 - 2\ell \geq 1$, since the length of the outer face is at most six. □

Notice that among above three rules, all vertices do not send any units of charge. Hence, the charge of all vertices is also non-negative. In consonance with the definition of discharging method, the total sum before and after charging should keep unchanged, which contributes to the existence of a face $f \neq f_0$ whose final charge is strictly negative. From rule (1) informed that f will send at most one unit to each incident vertex. The final charge is $3\ell - 12 - \ell = 2\ell - 12$, which is, as discussed, smaller than 0. It follows that the length of f is at most five. Moreover, if f has length exactly five, its initial charge is $3\ell - 12 = 3$, then f has to have at least four incident vertices. And all these vertices cannot be on C and have degree two. Otherwise, f will send nothing to the ends of the common subpath of f and f_0. As a deduction, the vertices of f structure a tetragram or a pentagram. □

Lemma 4.3. Every triangle-free plane graph G of minimum degree at least three has a safe tetragram, a safe pentagram or a safe hexagram. [3]

Proof. Let G as stated.

Claim 4. *Suppose that (v_1, v_2, v_3, v_4) is a tetragram in G which bounds a face, then one of tetragrams (v_1, v_2, v_3, v_4), (v_2, v_3, v_4, v_1) is safe.*

Proof. Prove by contradiction: assume that both of tetragrams are unsafe, which implies that there is no pair of diagonally opposite vertices.

(1) ***v_1 and v_3 is not the pair diagonally opposite vertices.*** So there is a path from v_1 to v_3 that is not part of graph G of length at three. See the case in **Figure 13**.

(2) ***v_2 and v_4 is not the pair diagonally opposite vertices.*** Then there must exist a path from v_2 to v_4 of length at most three.

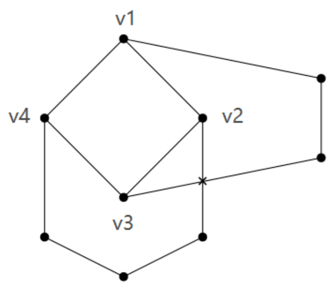

Figure 13: 4-face (v_1, v_2, v_3, v_4)

Since the tetragram (v_1, v_2, v_3, v_4) is a 4-face, the path has to be outside of the face, which results that there is a cross among two paths which is a contradiction to the planarity.

□

Therefore, we may assume that G has no 4-face. It follows that every 4-cycle is separating. If G features a separating cycle of length at the most five, then select the separating cycle C_1 in order that the bounded disk is smallest possible and let G_1 be the subgraph of G consisting of all vertices and edges drawn in the closed disk bounded by C_1. If G has no separating cycle of length at most five, then let $G_1 := G$ and let C_1 be a facial cycle of G of length at the most five. In addition, we reform the G that C_1 is the outer face.

Lemma 4.4. The minimum degree of G is at least three, then there exists a facial cycle C_1 of G of length at most five.

Proof. By contradiction, assume there is no face of length at most 5, which means every face has length at least 6. In this case, we have every vertex degree at least 3, so we have $e \geq \frac{3}{2} \cdot v \iff v \leq \frac{2}{3} \cdot e$ in view of the fact that each vertex contributes at least 3 edges, but we have counted each edge twice, so have to divide by 2. Similarly, since every face is of length at least 6, so $e \geq \frac{6}{2} \cdot f \iff f \leq \frac{1}{3} \cdot e$. Supposing that we plug these into Euler's formula, we get:

$$2 = v + f - e \leq \frac{2}{3}e + \frac{1}{3}e - e = 0$$

This is a contradiction. $\square$

Observe that no separating cycle of length at most five can be contained in G_1 for the reason that otherwise we can choose that smaller cycle to be C_1. Hence, there is no 4-cycle in G_1 except possibly C_1, because

- it cannot be a 4-face, as proved in **Claim 4**,
- there is no 4-separating cycle as mentioned.

A step further, we can define a subgraph G_2 of G_1 and the facial cycle C_2 of G_2 in this way: If there is a separating cycle of length six in G_2, then assign C_2 to be the cycle in which fewest vertices and edges are contained. G_2 is then the subgraph of all vertices and edges in which the cycle C_2 bounds. As analog to the case of G_1, there is no separating cycle of length at most six in G_2. Meanwhile, we've assumed that no 4-face is permitted, it leads that every cycle of length at most six in G_2 bounds a face. What's more, C_2 is a induced cycle from G_2. If it would not be the case, namely there would be a chord inside C_2, we could then choose a smaller cycle, which is a contradiction to the minimality.

We can apply **Lemma 4.2.** on graph G_2 and the facial cycle C_2 and conclude that G_2 has a pentagram $(v_1, v_2, v_3, v_4, v_5)$ with $v_1, v_2, v_3, v_4 \notin V(C_2)$.

(1) ***At least one of pentagrams (v_1, v_2, v_3, v_4, v_5) and (v_4, v_3, v_2, v_1, v_5) is safe.*** Then the lemma holds.

(2) ***None of them is safe.*** $\forall i \in \{1, 2, 3, 4\}$, let x_i be the neighbor of v_i other than v_{i-1} and v_{i+1}, where $v_0 = v_5$. Notice that x_1, x_2, x_3, x_4 belong to G_2 because otherwise it might create crosses between the incident edges with neighbors and C_2, and they are pairwise distinct and non-adjacent, since

   - G_2 is triangle-free,
   - no separating cycle at most 5,
   - no 4-cycle except C_2.

   It results $|\{x_1, x_2, x_3, x_4\} \cap V(C_2)| \leq 3$.

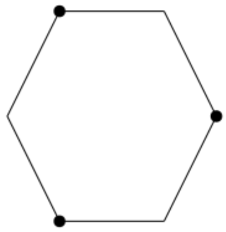

Figure 14: The number of intersection ≤ 3

If the number of intersection among these above two sets would be 4, then two of them have to be adjacent, which is a contradiction.

Lemma 4.5. If $v_5 \in V(C_2)$, then $\{x_1, x_2, x_3, x_4\} \cap V(C_2) = \emptyset$.

Proof. We assume that at least one of x_3 and x_4 is not on C_2.

Case 1: *x_4 is not on C_2*. Suppose that x_3 lies on C_2.

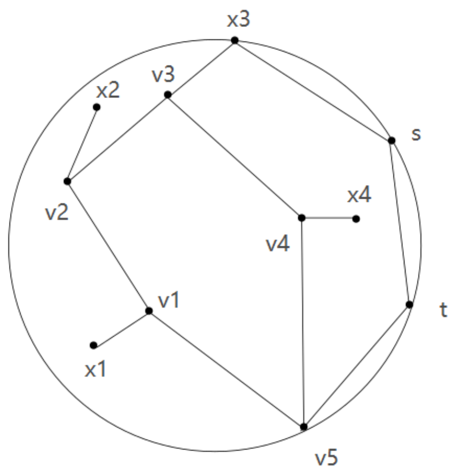

Figure 15: x_4 not on C_2, x_3 on C_2

It leads to a contradiction to the minimality of cycle C_2, for the reason that there is a smaller 4-separating cycle $x_3v_3v_4v_5$ or 5-separating cycle $x_3v_3v_4v_5s$ or 6-separating cycle $x_3v_3v_4v_5st$ containing x_4. Observe that if x_2 lies on C_2, it's analog to the former case when x_3 is on C_2. May assume now that x_1 lies on C_2:

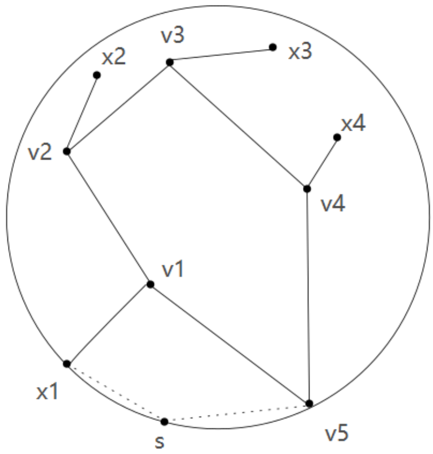

Figure 16: x_4 not on C_2, x_1 on C_2

It results also a contradiction to the minimality of C_2, because it's required to have at least one extra vertex on C_2 between x_1 and v_5, since no triangle in planar graph is permitted. In consequence, we find a smaller separating cycle which just visits v_1 with path $x_1 s v_5$ instead of visiting s with path $x_1 v_1 v_5$.

Case 2: x_3 *is not on* C_2. We may assume that x_4 is on C_2, otherwise it belongs to case 1.

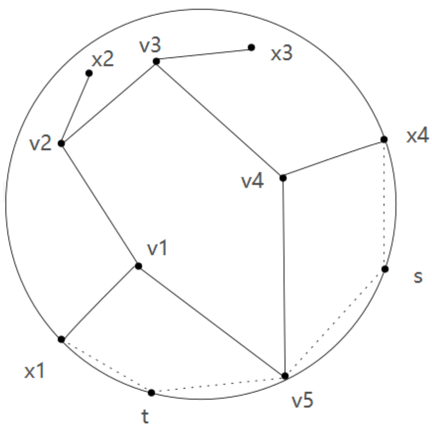

Figure 17: x_3 not on C_2, x_1 on C_2

All cases under this circumstance are totally similar to the case 1.

□

Since the pentagram $(v_1, v_2, v_3, v_4, v_5)$ is not safe, we know that, according to the definition of safety, there exists a pair of vertices x and y so that $\{x, y\} = \{x_3, x_4\}$ or

$\{x, y\} = \{x_2, v_5\}$ and a path P in $G \backslash \{v_1, v_2, v_3, v_4\}$ of length at most three ended with x and y, which either

- is from x_2 to v_5, or
- has length exactly three from x_3 to x_4 or its completion via the path $x_3v_3v_4x_4$ doesn't result in a facial cycle of length five in G.

If $\{x, y\} = \{x_3, x_4\}$, let Q be the path $x_3v_3v_4x_4$; otherwise $x_2v_2v_1v_5$. Then we have following cases:

Case 1: $P \cup Q$ ***bounds a facial cycle in*** G. Thus, we know that $\{x, y\} = \{x_3, x_4\}$ and the length of path P is exactly three. Suppose $P \cup Q$ is listed in this order $x_3v_3v_4x_4ab$.

Claim 5. *$(x_4, v_4, v_3, x_3, a, b)$ is a safe hexagram.*

Proof. By contradiction: there would be a path from x_4 to v_3 of length at most three that is not part of this hexagram. Assume that this path would be $x_4u_1v_3$ or $x_4u_1u_2v_3$, where $u_1, u_2 \neq v_4$.

- ***The path is*** $x_4u_1v_3$**.** Note that v_3 has degree exactly three according to the definition of pentagram and its neighbors are v_2, v_4, x_3. In addition, v_2 has also degree exactly three and has neighbors v_1, x_2, v_3. So u_1 can't be v_2. What's more, x_3, x_4 are non-adjacent. It follows that this case is not possible.
- ***The path is*** $x_4u_1u_2v_3$**.**

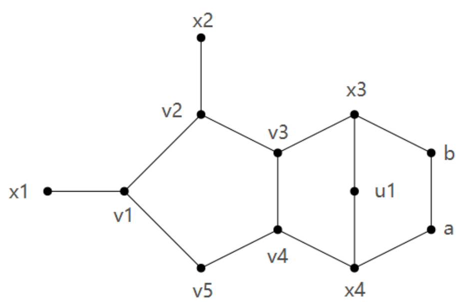

Figure 18: Path is $x_4u_1u_2v_3$.

Then $u_2 = x_3$. As discussed in **Lemma 4.5.**, at most one of x_3, x_4 can lie on C_2. Therefore, $x_4u_1u_2(x_3)v_3v_4$ would form a separating cycle of length five, which is a contradiction.

□

Case 2: $P \cup Q$ ***is a separating cycle in*** G. It results that $P \cup Q$ is not a subgraph of G_2, for every cycle of length at most six in G_2 bounds a face.

Claim 6. *At most one of $\{x, y\}$ can lie on C_2.*

Proof. Let's recall the following two cases:

(1) $\{x, y\} = \{x_2, v_5\}$: As pointed out in the **Lemma 4.5.**, if $v_5 \in V(C_2)$, x_2 can not belong to C_2.

(2) $\{x, y\} = \{x_3, x_4\}$: As mentioned in the **Lemma 4.5.** that at least one of x_3 and x_4 is not on C_2.

□

Thus, we may assume that $x_4 \in C_2$ and $x_3 \notin C_2$. The other case is similar. Meanwhile, a subpath R of $P \cup Q$ of length exactly four joins $w_1, w_4 \in V(C_2)$.

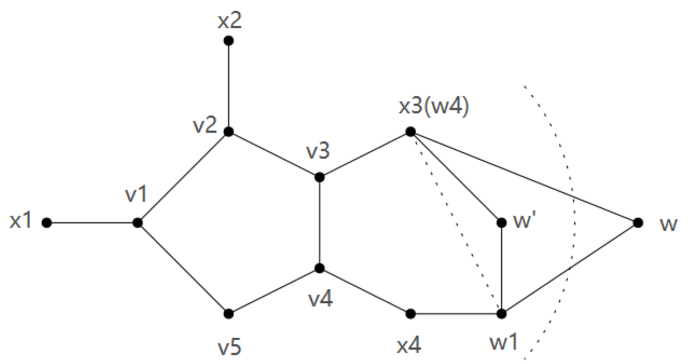

Figure 19: $P \cup Q$ is a separating cycle

Observe that the vertex $w \in (P \cup Q) \backslash V(G_2)$ is adjacent to w_1 and w_4. Let's take care of the position of w: assume firstly that $w \notin V(G_1)$, which implies that $w_1, w_4 \in V(C_1)$ because $w_1, w_4 \in V(C_2)$. Otherwise, edges ww_1 and ww_4 will build crosses with C_1. From the **Figure 19.** informed that w_1 and w_4 cannot be adjacent on the grounds that if this would not be the case, w_1w_4w will form a triangle, which is a contradiction. Thereby, there is a common neighbor of w_1 and w_4 in C_2 which can replace w. So we suppose that $w \in V(G_1)$.

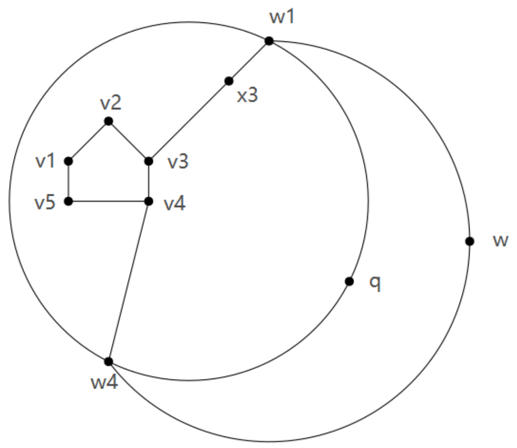

Figure 20: C_2: w_1, w_4 has a common neighbor q

Provided that there is common neighbor q of w_1, w_4 in C_2, call it q as in **Figure 20**. Then $w_1 q w_4 w$ would form a 4-cycle which is a contradiction because as mentioned before there is no 4-cycle in G_1 except possibly C_1. Hence, we conclude that there are two vertices w_2, w_3 between w_1 and w_4 as in **Figure 21**.

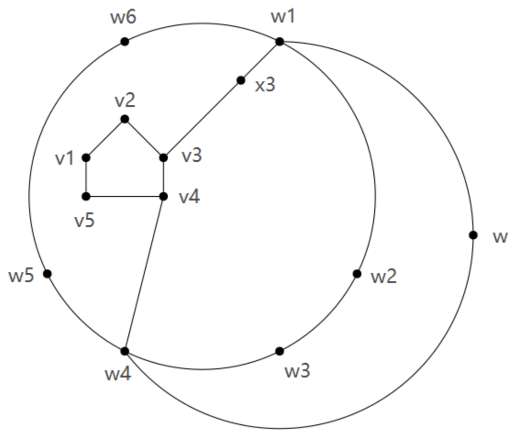

Figure 21: Between w_1, w_4 there are two vertices $w_2 w_3$ on C_2

Let $w_1 w_2 w_3 w_4 w_5 w_6$ be the vertices of C_2. What's more, the cycle bounded by $w_1 w_2 w_3 w_4 w$ has to be a 5-face. Otherwise, it would be a separating cycle, which is a contradiction to the minimality of C_1. Notice that inside $P \cup Q$ the only vertex w is not inside C_2. However, inside C_2, there are two more vertices w_5, w_6 which are not inside $P \cup Q$. So we should have chosen $P \cup Q$ instead of C_2 because of the minimality of C_2. which is a contradiction.

$\square$

Theorem 4.6 (Grötzsch's theorem)**.** Every triangle-free planar graph is 3-colorable. [4]

Proof. Let G as stated to be a triangle-free graph and we'll prove this theorem by induction.
Case 1: If the minimum degree is smaller than three, which means it can be one or two.

- if the minimum degree is one: search for vertices $\forall v \in V = \{v|deg(v) = 1\}$ and remove them from G to get a new graph $G^{'}$. If the minimum degree of G is two, then consider the case 1.2. Otherwise, we can apply the algorithm on $G^{'}$ and color $G^{'}$. Lastly, we can assign randomly a color different from the color of neighbor $\forall u \in \{u \mid u$ is the neighbor of $v, \forall v \in V\}$ to v.

- if the minimum degree is two: assume the vertex is $v \in V(G)$ and its neighbors $u, w \in V(G)$. Analogous to the former case, we remove the vertex v from G to obtain the new graph $G^{'}$. Suppose the coloring of u, w is c_1 and c_2. If $c_1 = c_2$, then we can color v randomly with a color different from $c_1(c_2)$. But if $c_1 \neq c_2$, we can accordingly assign c_3 to v.

Case 2: If every vertex v has degree at least 3, we can apply **Lemma 4.3.** on it. Thus, the theorem follows the induction on $G \backslash v$ by identifying pairs diagonally opposite vertices. And the way of coloring is already shown in the third section **Safety of multigram**: reconstruction step.

□

5 Implementation

5.1 Naive polynomial-time implementation

Description. The brute force method is that we start with a vertex $v \in V$ and just go through all its adjacent vertices which should be pairwise distinct. In this step, we will use multiple inner for-loops to achieve that. Notice that the case pentagram is kind of special. We need to check additionally whether v_1, v_2, v_3, v_4 have degree exactly three. Once we detect a multigram, then we should check the condition whether the found multigram is safe or not. [2]

- Found multigram is tetragram/hexagram. We can use the same way as above by using two for-loops to go through all neighbors and neighbors of neighbors so that we can check that there is no path of length at most three from v_1 to v_3 which is not part of the tetragram/hexagram. This can be done in $\mathcal{O}(m^2)$ time, where $m = |E|$.

- Found multigram is pentagram. As mentioned in the former case, we can also use two for-loops to check whether there is no path in $G \backslash \{v_1, v_2, v_3, v_4\}$ of length at most three from x_2 to v_5, and every path in $G \backslash \{v_1, v_2, v_3, v_4\}$ of length at most three from x_3 to x_4 has length exactly two. This can be done in $\mathcal{O}(m^2)$ time as well.

[2] The entire code can be found here: https://github.com/qiaw99/3-color-linear-time

□

Algorithm 1 The naive algorithm to detect multigrams and reduce the size of graph

Input: A triangle-free planar graph

```
adj := Adjacency list
for v ∈ V do
    for s ∈ adj[v] do
        for t ∈ adj[s] do
            for u ∈ adj[t] do
                                                    // Detect tetragrams
                if u ∈ adj[v] then
                    if the detected tetragram is safe then
                        Get the right pair of diagonally opposite vertices and identify
                          them.
                    end
                end
                else
                    for x ∈ adj[u] do
                                                    // Detect pentagrams
                        if x ∈ adj[u] then
                            if the detected pentagram is safe then
                                Get the right pair of diagonally opposite vertices and iden-
                                  tify them.
                            end
                        end
                        else
                            for y ∈ adj[x] do
                                                    // Detect hexagrams
                                if y ∈ adj[x] then
                                    if the detected hexagram is safe then
                                        Get the right pair of diagonally opposite vertices
                                          and identify them.
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
```

Note that according to the previous description, this brute force algorithm for detecting multigrams and identifying vertices will take $\mathcal{O}(m^8)$ time.

We will give a small example how the program executes the given input graph step by step. The followings **Figures 22 - 25** are the input graph and its result. The input graph contains two reducible multigrams: a tetragram and a pentagram.

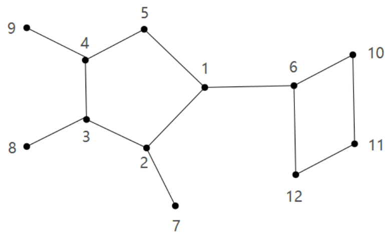

Figure 22: The input graph

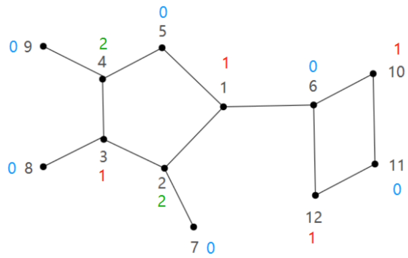

Figure 23: The proper coloring

```
The info of input graph:
Tetragram:
[6, 10, 11, 12]
Pentagram:
[1, 2, 3, 4, 5]
Hexagram:
Number of tetragram: 1
Number of pentagram: 1
Number of hexagram: 0
Remaining #vertex: 12

Identified vertices: [6, 11] in 0-th iteration
Tetragram:
Pentagram:
[1, 2, 3, 4, 5]
Hexagram:
Number of tetragram: 0
Number of pentagram: 1
Number of hexagram: 0
Remaining #vertex: 11

Identified vertices: [7, 5] in 1-th iteration
Identified vertices: [8, 9] in 2-th iteration
Tetragram:
Pentagram:
Hexagram:
Number of tetragram: 0
Number of pentagram: 0
Number of hexagram: 0
Remaining #vertex: 5
```

Figure 24: Detect multigrams and identify vertices [a]

```
Vertex 0 ---> Color 0
Vertex 1 ---> Color 0
Vertex 2 ---> Color 0
Vertex 3 ---> Color 0
Vertex 4 ---> Color 0
Vertex 5 ---> Color 0
Vertex 6 ---> Color 0
Vertex 7 ---> Color 0
Vertex 8 ---> Color 0
Vertex 9 ---> Color 0
Vertex 10 ---> Color 1
Vertex 11 ---> Color 0
Vertex 12 ---> Color 1
************************
Vertex 0 ---> Color 0
Vertex 1 ---> Color 1
Vertex 2 ---> Color 2
Vertex 3 ---> Color 1
Vertex 4 ---> Color 2
Vertex 5 ---> Color 0
Vertex 6 ---> Color 0
Vertex 7 ---> Color 0
Vertex 8 ---> Color 0
Vertex 9 ---> Color 0
Vertex 10 ---> Color 1
Vertex 11 ---> Color 0
Vertex 12 ---> Color 1
************************
```

Figure 25: The corresponding coloring

[a] Implemented in Java (JDK 1.8)

5.2 Improvement to the Brute-Force algorithm

We can actually use some tricks to detect safe tetragrams and hexagrams more efficiently. Recall that tetragrams or hexagrams are called safe, if every path in G of length at most three with ends v_1 and v_3 is a subgraph of cycle $v_1 v_2 ... v_k$ (where $k = 4$ or 6).

- Tetragram. We can use two loops to go through all edges $e_1, e_2 \in E$, where $e_1 \neq e_2$. As the first step, we have to check whether e_1, e_2 forms a cycle. Next, we should check the condition of safety by using BFS. Observe that the running time of this improved algorithm for detecting tetragrams is $\mathcal{O}(m^3)$, since BFS takes just $\mathcal{O}(m)$ time.

Algorithm 2 Improvements to detect tetragrams

Input: A triangle-free planar graph
adj := Adjacency list
for *every* $(x, y) \in E$ **do**
  for *every* $(s, t) \in E$ *with* $(s, t) \neq (x, y)$ **do**
    if *(*$x \in adj[s]$ *and* $y \in adj[t]$*) or (*$x \in adj[t]$ *and* $y \in adj[s]$*)* **then**
      Do BFS to get all paths of length at most three from a to b, where $a, b \in \{x, y, s, t\}$ and a, b are distinct
      if $\exists a, b$: *all path from a to b of length at most three are part of tetragram* **then**
        (a, c, b, d) is a safe tetragram, where $c, d \in \{x, y, s, t\} \backslash \{a, b\}$
      end
    end
  end
end

- Hexagram. The efficient way to detect hexagrams is analog to the previous case. We will use three loops to go through all edges $e_1, e_2, e_3 \in E$, where $e_1 \neq e_2 \neq e_3$. Firstly, we check if e_1, e_2, e_3 builds a cycle. Then we have to take care of the safety of the hexagram by using BFS as well. Totally, the running time is $\mathcal{O}(m^4)$.

- Pentagram. We use two loops to go through all edges $e_1, e_2 \in E$, where $e_1 \neq e_2$ and another loop to go through all remaining vertices. Then, we apply BFS to check wether there is a path in $G \backslash \{v_1, v_2, v_3, v_4\}$ of length at most three from x_2 to v_5 and every path in $G \backslash \{v_1, v_2, v_3, v_4\}$ of length at most three has length exactly two. Therefore, the running time is $\mathcal{O}(m^4)$.

So the running time can be improved to $\mathcal{O}(m^4)$ in the worst case.

5.3 Reconstruction step

The algorithm will keep detecting all reducible multigrams until there is no multigram within the graph G' obtaining by identifying vertices in G. At this moment, the graph is easy to be colored. While identifying vertices, we will maintain two lists: one is to store the graph each time and the other is to store the pairs of identified vertices, with which we can reconstruct the graph coloring according to the subsection **3.2.**

6 Improvements from the original paper

As explained in the second section that in our algorithm, we will find some reducible multigrams to cut down the size of graph G. We've seen the algorithms in the fifth section and know that safety is not strong enough to get a linear-time algorithm. The difficulty now is how to efficiently check whether the found multigrams are safe, for which we will introduce another concept called *security* that allows us testing safety in constant time. Before that, let's give a few more definitions first.

6.1 Definitions

Definition 6.1. A *monogram* in the graph G is the one-vertex sequence consisting of a vertex $v \in V(G)$ and has degree at most two. [3]

Let G be a triangle-free planar graph, $k \in \{1,4,5,6\}$ and $\gamma = (v_1, ..., v_k)$ a mono-, tetra-, penta- or hexagram in G. Let C be the subgraph of G.

Definition 6.2. A vertex $v \in V(G)$ is *big*, if it has degree at least 60 and *small* otherwise. [3]

Definition 6.3. A vertex $v \in V(G)$ is called *C-admissible*, if v is small and $v \notin C$; otherwise called *C-forbidden*. [3]

Definition 6.4. A pentagram $(v_1, v_2, v_3, v_4, v_5)$ is a *decagram* if v_5 has degree exactly three as well. A tetragram (v_1, v_2, v_3, v_4) is *octagram* if all vertices have degree three. [3]

Furthermore, we extend the concept multigram as follows: a *multigram* is a monogram, tetragram, pentagram, hexagram, octagram or decagram.

Definition 6.5. Let γ be a multigram $(v_1, v_2, ..., v_k)$. The vertex v_1 is called *pivot* of multigram. [3]

6.2 Security of multigrams

Next, we'll explain what's the meaning of C-secure and define a smaller graph G' which is called *γ-reduction of G*.

(1) γ ***is a monogram***. We define it to be always safe. γ is C-secure, if $v_1 \notin V(C)$ and $G' := G \backslash v_1$.

(2) γ ***is a tetragram***.

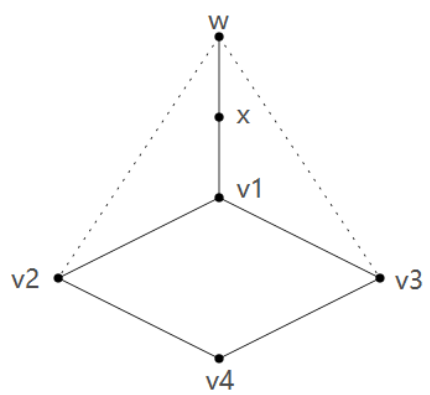

Figure 26: C-secure tetragram

γ is C-secure if

- γ is safe,
- v_1 is C-admissible and has degree exactly three,
- letting x be the neighbor of v_1 other than v_2 and v_4, the vertex x is C-admissible,
- either
  * v_3 is C-admissible, or
  * every neighbor w of x is C-admissible or belongs to a 4-face incident with the edge v_1x (either v_1v_2wx or v_1v_4wx).

$G^{'}$ be the graph obtained by identifying vertices v_1 and v_3 and delete one edge from each f the two pairs of parallel edges.

(3) γ ***is an octagram***. We define it to be always safe as well. γ is C-secure, if v_1, v_2, v_3, v_4 are C-admissible. $G^{'} := G \backslash \{v_1, v_2, v_3, v_4\}$.

(4) γ ***is a decagram***. $\forall i \in \{1, 2, 3, 4\}$, x_i is the neighbor of v_i other than v_{i-1} and v_{i+1}, where $v_0 = v_5$. The decagram γ is *safe*, if

- x_1, x_3 are distinct and non-adjacent,
- there is no path of length two between them.

The decagram γ is $C-$secure, if

- γ is safe,
- $v_1, v_2, v_3, v_4, v_5, x_1, x_3$ are C-admissible.

$G^{'}$ is obtained from $G \backslash \{v_1, v_2, v_3, v_4, v_5\}$ by adding the edge x_1x_3.

(5) γ ***is a pentagram***. $\forall i \in \{1, 2, 3, 4\}$, x_i is the neighbor of v_i other than v_{i-1} and v_{i+1}, where $v_0 = v_5$. γ is C-secure, if

- $v_1, v_2, v_3, v_4, v_5, x_1, x_2, x_3, x_4$ are C-admissible,
- either v_5 or x_2 has no C-forbidden neighbor,
- either x_3 or x_4 has no C-forbidden neighbor.

$G^{'}$ is obtained from $G \backslash \{v_1, v_2, v_3, v_4\}$ by identifying v_5 and x_2; x_3 and x_4. If x_3 and x_4 should have a common neighbor, then delete one of the parallel edges.

(6) γ ***is a hexagram***. γ is C-secure, if

- v_1, v_3, v_6 are C-admissible,
- v_1 has degree exactly three,
- the neighbor of v_1 other than v_2 or v_6 is C-admissible.

$G^{'}$ be the graph obtained by identifying vertices v_1 and v_3 and delete one edge from each f the two pairs of parallel edges.

Definition 6.6. A graph G is *null* if $V(G) = \emptyset$. [11]

Definition 6.7. A multigram is *secure*, if it is K_0-secure, where K_0 is the null graph. [3]

6.3 Constant-time operations

Note that the following lemmas will not be proved in this paper, see proofs from the original paper.

Lemma 6.1. Let G be a triangle-free plane graph, let γ be a safe multigram in G, and let $G^{'}$ be the γ-reduction of G. Then $G^{'}$ is triangle-free, and every 3-coloring of $G^{'}$ can be converted to a 3-coloring of G in constant time. Moreover, if γ is secure, then $G^{'}$ can be regarded as having been obtained from G by deleting at most 126 edges, adding at most 116 edges, and deleting at least one isolated vertex.[3]

Definition 6.8. *Two small vertices $u, v \in V(G)$ are close* if either there is a path of length at most four between u and v consisting of small vertices, or a facial cycle of length at most six contains both u and v. *A vertex u is close to an edge e* if both u and e belong to the facial walk of the same face and the distance between u and and one end of e in this facial walk is at most two. [3]

Claim 7. *For every vertex v there are at most $1 + 4 \cdot 59 + 59^2 + 59^3 + 59^4$ vertices that are close to v, and for every edge e, there are at most 10 vertices that are close to e. [3]*

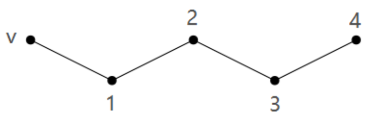

Figure 27: Case 1

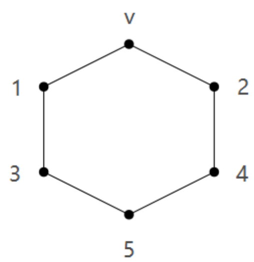

Figure 28: Case 2

Proof. Observe that in the **Figure 27**, vertex 1 can have at most 59 neighbors, vertex 2 can have at most 59^2 neighbors, vertex 3 can have at most 59^3 neighbors and vertex 4 can have at most 59^4 neighbors. In the **Figure 28**, vertex 1 and 2 can totally have at most 59 neighbors, while vertex 3 and 4 each can have 59 neighbors. But vertex 5 can have only one vertex. So all things considered there are at most $1 + 4 \cdot 59 + 59^2 + 59^3 + 59^4$ vertices that are close to v.

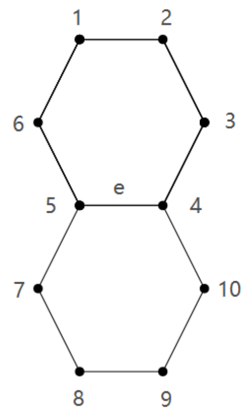

Figure 29: Vertices close to edge e

Note that both v and e have to belong to the same walk of the same face. In addition, there are at most two faces that incident with edge e, which follows that there are at most 10 vertices that are close to edge e. □

Lemma 6.2. Given a triangle-free plane graph G and a vertex $v \in V(G)$, it can be decided in constant time whether G has a secure multigram with pivot v. [3]

Lemma 6.3. Let G and $G^{'}$ be triangle-free plane graphs, such that for some pair of non-adjacent vertices $u, v \in V(G)$ the graph $G^{'}$is obtained from G by adding the edge uv. Let γ be a secure multigram in exactly one of the graphs G, $G^{'}$. Then the pivot of γ is close to u or v in G, or to the edge uv in $G^{'}$. [3]

Theorem 6.4. Every non-null triangle-free planar graph has a secure multigram.

Proof. See the proof from the original paper. [3] □

6.4 The algorithm

Now, let's focus on the linear-time algorithm with running time $\mathcal{O}(|V(G)|)$: we may assume that the input graph is triangle-free, because we can use the algorithm which was designed by Hopcroft and Tarjan [5] to test planarity in $\mathcal{O}(n)$ time by *path addition method*. So it holds our algorithm's running time in linear-time.

Algorithm 3 3-coloring in triangle-free planar graph

```
Input: A triangle-free planar graph
Output: A 3-coloring of G
L := {v ∈ V(G)|deg(v) ≤ 3}                                  // L is a list
temp := G                                     // Store the original graph G
for v ∈ L do
    Remove v from L
    if G has a multigram with pivot v then
        γ := such safe multigram
        G' := γ-reduction of G
        for every edge uv deleted or added during construction of G' do
            Add to L all vertices that are close to u or v or to the edge uv in G or G'
        end
        G := G'                                    // To reduce the size of G
    end
    else
        continue to next iteration
    end
end
while G ≠ temp do
    Color the graph G                                   // Reconstruction step
    Convert the 3-coloring of G to the graph G' before the γ-reduction
    G := G'
end
```

Description. We initialize L as the list of vertices with degree at most three. During the execution of the algorithm, L includes all pivots of all secure multigrams. The algorithm works as follows: remove randomly a vertex v from L and check whether there is a multigram whose pivot is v, which can be done in constant time according to **Lemma 6.2.** If there is no such multigram, then we go to the next iteration. Otherwise, let γ be such multigram and $G^{'}$ be the γ-reduction of G, which can be done in constant time according to **Lemma 6.1. Lemma 6.3.** can guarantee that L includes all pivots of all secure multigrams. After we get the coloring of $G^{'}$, it can be converted to the coloring of G. And we'll do this until we get the coloring of the original input graph G. Notice that the number of vertices added to L is proportional to the number of vertices removed from G. Hence, the running time is $\mathcal{O}(|V(G)|)$, as desired. □

Algorithm 4 3-coloring in triangle-free planar graph with coloring constraint

Input: A triangle-free plane graph G, a facial cycle C in G of length at most five, and a proper 3-coloring ϕ of C.

Output: A proper 3-coloring of G whose restriction to $V(C)$ is equal to ϕ.

Description. The description is exactly the same, except that we replace "secure" by "C-secure" and appeal to **Lemma 7.1** rather than **Theorem 6.4.** And its running time is still $\mathcal{O}(|V(G)|)$.[3] □

7 Proof of correctness

Definition 7.1. f *is opposite to* xy, if xy is an edge in a planar graph, and f is a face of G incident with y but not with the edge xy. Note that this definition is not symmetric: f is opposite to xy, but f is *not* opposite to yx. [3]

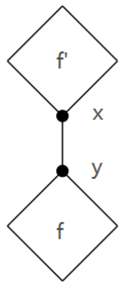

Figure 30: xy-opposite and yx-opposite faces

Lemma 7.1. Let G be a connected triangle-free planar graph and let f_0 be its outer face. Assume that f_0 is bounded by a cycle C of length at most six, $V(G) \neq V(C)$, and if C has length six, then $|V(G) - V(C)| \geq 2$. Then G contains a C-secure multigram. [3]

Proof. By contradiction, presuming that G is a counterexample with $E(G)$ minimum, namely G doesn't contain any C-secure multigram.

Corollary 7.1.1. If $K \neq C$ is a cycle in G of length at most six, then K bounds a face, or K has length six and the open disk bounded by K contains at most one vertex. [3]

We conclude that C is induced and every tetragram is safe the reason is that from **Corollary 7.1.1.**, we know that a cycle in G of length four will bound a face. Hence, according to **Claim 4**, those tetragram is always safe.

Next, we assign the following charges to the vertices and faces of G. Primitively, a vertex $v \in V(G)$ will obtain a charge of

$$\begin{cases} 9\,deg(v) - 36, & v \notin V(C) \\ 8\,deg(v) - 19, & otherwise \end{cases}$$

and a face will take a charge of

$$\begin{cases} 0, & f = f_0 \\ 9\ell - 36, & f \neq f_0 \text{ and } f \text{ has length } \ell \end{cases}$$

Claim 8. *The sum of charges is negative.*

Proof.

$$
\begin{aligned}
sum\ of\ charges\ &= \sum_{v \notin V(C)} 9\big(deg(v) - 4\big) + \sum_{v \in V(C)} \big(8\ deg(v) - 19\big) + \sum_{f \neq f_0} 9\big(size(f) - 4\big) \\
&= \underbrace{\sum_{v \in V(G)} 9\big(deg(v) - 4\big) - \sum_{v \in V(C)} 9\big(deg(v) - 4\big)}_{\sum_{v \notin V(C)} 9\big(deg(v)-4\big)} + \sum_{v \in V(C)} \big(8\ deg(v) - 19\big) + \sum_{f \neq f_0} 9\big(size(f) - 4\big) \\
&= \sum_{v \in V(G)} 9\big(deg(v) - 4\big) + \sum_{f \neq f_0} 9\big(size(f) - 4\big) - \sum_{v \in V(C)} \big(deg(v) - 17\big) \\
&= \sum_{v \in V(G)} 9\big(deg(v) - 4\big) + \sum_{f \neq f_0} 9\big(size(f) - 4\big) - \sum_{v \in V(C)} deg(v) + 17\,|V| \\
&= \sum_{v \in V(G)} 9\big(deg(v) - 4\big) + \sum_{f} 9\big(size(f) - 4\big) - 9\big(size(f) - 4\big) - \sum_{v \in V(C)} deg(v) + 17\,|V(C)| \\
&= \sum_{v \in V(G)} 9\big(deg(v) - 4\big) + \sum_{f} 9\big(size(f) - 4\big) - 9\,|V(C)| + 36 - \sum_{v \in V(C)} deg(v) + 17\,|V(C)| \\
&= \underbrace{18\,|E| - 36\,|V(G)| + 18\,|E| - 36\,|F|}_{\text{apply Euler's formula here: v + f - e = -2}} - \sum_{v \in V(C)} deg(v) + 8\,|V(C)| + 36 \\
&= -36 \cdot 2 - \sum_{v \in V(C)} deg(v) + 8\,|V(C)| + 36 \\
&= 8\,|V(C)| - 36 - \sum_{v \in V(C)} deg(v) \\
&\leq 8\,|V(C)| - 2\,|V(C)| - 1 - 36 \\
&\leq -1\ (since\ the\ length\ of\ C\ is\ at\ most\ 6)
\end{aligned}
$$

The reason of first inequality is that all vertices of cycle C have at least degree two and there is at least one vertex of C whose degree is at least three, which follows $\sum_{v \in V(C)} deg(v) \leq 2\,|V(C)| + 1$.

□

Definition 7.2. Given the graph G, two edges $e_1 = v_1v_2, e_2 = v_3v_4$, where $v_1, v_2, v_3, v_4 \in V(G)$ and $e_1, e_2 \in E(G),\ e_1 \neq e_2$, are *consecutive*, if $v_2 = v_3$.

Definition 7.3. Assume $f \neq f_0$ is a face contained in G incident with a vertex $v \in V(C)$. Provided that there exist two consecutive edges within the boundary of f such both are incident with v and neither belongs to C, then we are saying that f is a $v - interior\ face$. [3]

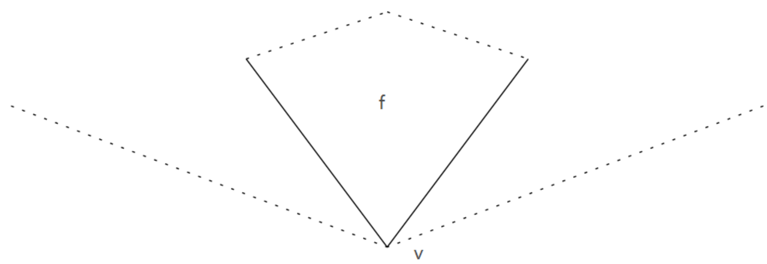

Figure 31: v-interior face f

Corollary 7.1.2. If at least k vertices of C have degree at least three, then the sum of the charges is at most -k. [3]

We now redistribute the charges consistent with the subsequent rules. The rules are:[3]

(A) Every face other than f_0 sends three units of charge to every incident vertex v such that either $v \in V(C)$ and v has degree two in G, or $v \notin V(C)$ and v has degree exactly three.

(B) Every big vertex not on C sends three units to each incident face, and four units to each 4-face that shares an edge with C.

(C) Every vertex $v \in V(C)$ sends three units to every v-interior face.

(D) If $x \in V(G)$ is C-forbidden, and y is a C-admissible neighbor of x of degree three, then x sends three units to the unique face opposite to xy, and one unit to the face opposite to yz for every C-admissible neighbor z of y of degree three.

(E) Every C-forbidden vertex sends five units to every C-admissible neighbor of degree at least four.

(F) For every C-admissible vertex y of degree at least four that has a C-forbidden neighbor we select a C-forbidden neighbor x of y and let y send one unit to each face opposite to xy, and one unit to the face opposite to yz for every C-admissible neighbor z of y of degree three.

Since G is a counterexample of the lemma, G doesn't contain any C-secure multigram, which follows that every vertex $v \in V(G)$ has degree at least two and the vertices with degree two are on C.

Claim 9. *Every vertex $v \in V(G)$ with degree d has non-negative charge.*

Proof. We will observe the following cases:

Case 1.1: ***v is C-admissible and $d = 3$.*** At the beginning, v has the charge $9 \cdot 3 - 36 = -9$. Pursuant to the rule (A), v will get totally 9 units from three incident faces other than f_0, which results that the final charge of v is zero (non-negative).

Case 1.2.1: v ***is C-admissible,*** $d \geq 4$ ***and has no C-forbidden neighbor.*** The original charge of v is $9d - 36 \geq 0$. What's more, v doesn't send out any charges. So the final charge is still non-negative.

Case 1.2.2: v ***is C-admissible,*** $d \geq 4$ ***and has a C-forbidden neighbor.*** Let x be the C-forbidden neighbor of v. Informed from the rule (E) that v will receive five units from x. Meanwhile, v sends out at most (2d - 3) units according to the rule (F).

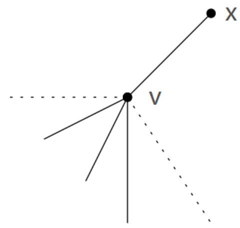

Figure 32: Case 1.2.2

Notice that v has remaining $(d-1)$ neighbors except x. In the worst case, all $(d-1)$ neighbors have degree exactly three. Thus, v will send out at most $(d-1)$ units to each face f that f is opposite to vz, $\forall z \in Z$, where Z is the set of all neighbors of v except x. Furthermore, observe from **Figure 32** that, $(d-1)$ edges(neighbors) can at most form $(d-2)$ xv-opposite faces. As a result, v sends out at most $(d-2+d-1) = 2d-3$ units.
$\longrightarrow$ The final charge of $v = 9d - 36 + 5 - (2d-3) = 7d - 28 \geq 0$ is also non-negative.

Case 2.1: v ***is big and*** $v \notin C$. As stated in the rule (B), v sends $3d$ units to all incident faces and at most 4·6 = 24 units to all 4-faces that share an edge of C, since C has length at most six. Additionally, v sends at most $5d$ units using the rule (D),

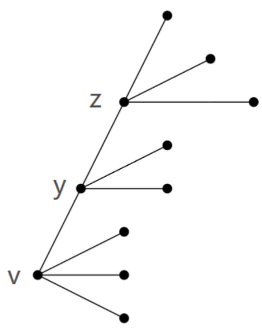

Figure 33: Case 2.1 using the rule (D)

The reason is that each neighbor y of v can have degree three and be C-admissible which can form at most two unique faces opposite to vy. In addition, every vertex

y can have at most three neighbors. Each neighbor z can also be C-admissible of degree three. Hence, there are at most three faces opposite to yz for each y. In conclusion, v sends out at most five units for each neighbor, which deduces v can send at most $5d$ units using rule (D). Or using rule (E), provided that all neighbors of v can be C-admissible and have degree at least 4. As a consequence, the final charge of $v = 9d - 36 - 3d - 24 - 5d = d - 60 \geq 0$, since v is big.

Case 2.2: $v \in C$. The charge of v at the beginning is $8d - 19$. Note that if $d = 2$, by the rule (A), v will receive three units. So we assume that is not the case. Otherwise, according to rule (A), v sends out $3(d-3)$ units by the rule (C), because v has two neighbors that are on C, which leads that the remaining $(d-2)$ neighbors can form $(d-3)$ v-interior faces. Besides, v sends $5(d-2)$ units using rule (D) or (E). The argument is similar to the former case 2.2: there are at most $(d-2)$ C-admissible neighbors of v. In consequence, the final charge of v = $8d - 19 - 3(d-3) - 5(d-2) = 0$.

□

Claim 10. *Every face of length $\ell \geq 6$ has non-negative final charge.*

Proof. Notice from rule (A) that each face $f \neq f_0$ can just send at most 3ℓ units. So the final charge = $9\ell - 36 - 3\ell = 6\ell - 36 \geq 0$. □

From the **Claim 9** we prove that there is a face $f \neq f_0$ in G of length at most five that has strictly negative final charge.

Corollary 7.1.3. No vertex incident with f has degree two. [3]

Proof. Prove using contradiction: there is a vertex v incident with f with degree exactly two. Thereby, v and the two incident edges are on C, which implies that there are at least two vertices of f are on C that will not receive any charges from f. Because of the fact that the face f has strictly negative final charge, we deduce that the length of f is four:

$$\begin{aligned} The\ final\ charge &= 9\ell - 36 - \big(1 \cdot 3 + (\ell - 2 - 1) \cdot 3\big) < 0 \\ &\Longrightarrow \ell < 5 \\ &\Longleftrightarrow \ell \leq 4 \end{aligned}$$

Let $u_1u_2u_3u_4$ be the bounded face, where $u_1u_2u_3$ are consecutive vertices of C and u_2 has degree exactly two. Meanwhile, $u_4 \notin C$, since C is induced. Then u_4 is small and hence C-admissible. The reason is that if u_4 would be big, by using rule (B), u_4 would send seven units to f. Furthermore, f has initial charge $9 \cdot 4 - 36 = 0$ and f sends three units to u_2 by rule (A). As a result, the final charge of f would be $0 - 3 + 7 = 4$, which is a contradiction. Let $C^{'}$ be the cycle obtained from C by replacing the vertex u_2 by u_4. Since we replace u_2 by u_4, it follows $|V(C^{'})| = |V(C)| \leq 6$. So the length constraint holds.

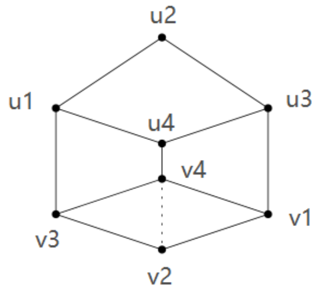

Figure 34: Cycle C and $C^{'}$

As above mentioned, u_4 has degree at least three, $C^{'}$ can't form a face, which follows from **Corollary 7.1.1** that $C^{'}$ has length six and contains exactly one vertex v_4. Next, we order the remaining three vertices on C as in **Figure 34** with $v_1v_2v_3$ so that the original cycle C is $u_1u_2u_3v_1v_2v_3$ and v_4 is adjacent to u_4, v_1, v_3. Observe that:

(1) (u_4, u_1, u_2, u_3) is safe, since there is not path at most three other than $u_4u_1u_2$ and $u_4u_3u_2$ that ends with u_2 and u_4.

(2) As proved above, u_4 is C-admissible and has degree exactly three.

(3) Notice from the figure that the neighbor v_4 of u_4 is not on C and has degree exactly three as well. It follows that v_4 is also C-admissible.

(4) v_4 belongs to the 4-face(u_4, u_1, v_3, v_2) incident with u_4v_4.

Resultantly, (u_4, u_1, u_2, u_3) is a C-secure tetragram, which is a contradiction to the supposition that there is no C-secure multigram in G. □

Definition 7.4. Let uv be the edge to which f is opposite. v is a *sink*, if v has degree three and both u and v are C-admissible. v is a *source*, if either $v \notin V(C)$ and v is big, or $v \in V(C)$ and f is v-interior. [3]

Observation 5. *v is **not** a source, then either v is small and $v \notin V(C)$; or $v \in V(C)$ and f is not v-interior.*

Observation 6. *The equivalences of definition of sink and source are stated as follows: [3]*

- *v is a sink $\Longleftrightarrow$ v has degree three and receives three units of charge from f by rule (A) and f does not receive three units by rule (D) from u.*
- *v is a source $\Longleftrightarrow$ v sends three units of charge to f by rule (B) or (C).*

Suppose s is the number of sources and t is the number of sinks. Then we have the initial charge of f $9 + 3s - 3t$, if the length of f is five; $3s - 3t$, if the length of f is four.

Case 1.1: ***f has length five and v_5 is C-admissible of degree three.*** Let $v_1v_2v_3v_4v_5$ be the cycle that bounds f. For f has strictly negative final charge in the end, there are at least four sinks. Hence, we may assume that v_1, v_2, v_3, v_4 are sinks. In other word,

they are C-admissible and have degree exactly three. In addition, $(v_1, v_2, v_3, v_4, v_5)$ is a pentagram according to the definition. Let x_i be the neighbor of v_i except v_{i-1} and v_{i+1}, $\forall i \in \{1,2,3,4\}$, where $v_0 = v_5$. It infers that x_1, x_2, x_3, x_4 are distinct and pairwise non-adjacent. The reason is as follows:
(Case 1) If x_i and x_{i+1} would be adjacent, it will create a C-secure tetragram with vertices v_i and v_{i+1}, which is a contradiction.
(Case 2) If x_i and x_{i+2} would be adjacent, from **Corollary 7.1.1.** that $x_i x_{i+2} v_{i+2} v_{i+1} v_i$ bounds a face. However, it contains a vertex x_{i+1}, which is a contradiction as well.
Notice from (Case 2) that there is no path from x_1 to x_3 with length two, then $(v_1, v_2, v_3, v_4, v_5)$ is a C-secure decagram.

Case 1.2: ***f has length five and v_5 is not C-admissible of degree three.*** If there is a path from x_1 to x_3 of length three, then consider the cycle $K = x_1 v_1 v_2 v_3 x_3 y$.

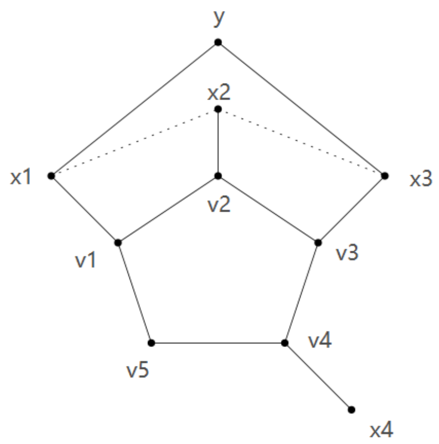

Figure 35: Case 1.2

Since the cycle K has length six and from **Corollary 7.1.1.** that K contains at most one vertex, it results that v_4 and v_5 are not inside K. Hence, either $y = x_2$ or x_2 is the vertex containing in the cycle K. Therefore, x_2 is adjacent to x_1 and x_3. In the former case, it's obvious true. In the latter case, we can prove it by contradiction: if x_2 is not adjacent to one of x_1 and x_3, it will form a triangle, which is against to the definition of planar graph. So in any case, x_2 is adjacent to x_1 and x_3, which is a contradiction to what we've proved in the case 1.1 that x_1, x_2, x_3 should be pairwise non-adjacent.

Consequently, from both cases above that v_5 is either not C-admissible or has degree at least four, which implies that v_5 is not a sink and then the final charge of f is at least $9 - 3 \cdot 4 = -3$. What's more. v_5 is not a source neither. If v_5 would be a source, then the final charge of f would be zero, which is a contradiction to the negative final charge of f. Thus, it deduces from **Observation 5.** that v_5 is C-admissible and hence has degree at least four. We may claim that the pentagram $(v_1, v_2, v_3, v_4, v_5)$ is safe:

- there is a path of length at most three in $G\backslash\{v_1, v_2, v_3, v_4\}$ that ends with x_2 and $v5$, which can be completed via a path $v_5v_1v_2$ to be a new cycle K.

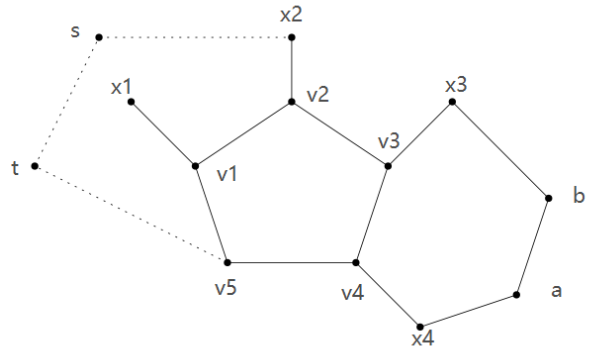

Figure 36: Safe pentagram

From **Corollary 7.1.1.** that K contains at most one vertex. On the grounds of this, x_1 is adjacent to x_2(the argument is the same as in the case 1.2), which is a contradiction.

- considering the path of length at most three from x_3 to x_4 that can be completed to a cycle $K^{'}$ ($x_3v_3v_4x_4ab$) by the path $x_4v_4v_3x_3$ supposing that the length of cycle is six. Observe that v_3, v_4 are sinks, so v_3 has degree exactly three and x_3, x_4 are C-admissible. Thus, the hexagram $(v_4, v_3, x_3, b, a, x_4)$ is C-secure, which is a contradiction. It follows that the path from x_3 to x_4 can only be two.

According to the above proof, we have shown that the pentagram $(v_1, v_2, v_3, v_4, v_5)$ is safe. By symmetry the pentagram $(v_4, v_3, v_2, v_1, v_5)$ is safe as well. Meanwhile, x_1, x_2, x_3, x_4 are C-admissible, for v_1, v_2, v_3, v_4 are sinks. Provided that the neighbor x_i of v_i for $i \in \{1, 2, 3, 4\}$ has a C-forbidden neighbor u, then f will receive one unit either from rule (D) if u has degree exactly three or (F) if u has degree at least four. Notice that v_5 has degree at least four and if v_5 has a C-forbidden neighbor, then f will receive one unit by rule (F). Note that initially, f has charge 3. Hence, at most two vertices among x_1, x_2, x_3, x_4, v_5 have a C-forbidden neighbor. In consequence, either $(v_1, v_2, v_3, v_4, v_5)$ or $(v_4, v_3, v_2, v_1, v_5)$ is a C-secure pentagram, which is a contradiction.

We've proved that f has then length exactly four:

Case 2: ***f has length four.*** Let v_1, v_2, v_3, v_4 be the incident vertices of f. Recall that every tetragram is safe and f has initial charge $3s - 3t$. We may assume that v_1 is a sink and v_3 is not a source. Consequently, $v_3 \in V(C)$ and f is not v-interior, for v_3 is not a source and (v_1, v_2, v_3, v_4) is not a C-secure tetragram (implies v_3 is not C-admissible). What's more, only one of edges v_2v_3, v_3v_4 is shared with C. And we may assume the latter edge is the case, which implies $v_2 \notin V(C)$. d that if v_2 is a sink, then the charge of f is at least -6, otherwise -3. Let v be the neighbor of v_1 different than v_2 and v_4:

  - ***v has no C-forbidden neighbor.*** (v_1, v_2, v_3, v_4) is then a C-secure tetragram, which is a contradiction.

- ***v* has a *C*-forbidden neighbor *u*.**
  - $u \notin V(C)$. Therefore, u is big and f receives four units from u, since f is a 4-face and shares an edge v_3v_4 with C by rule (B). At this moment, the charge of $f = -3 + 4 = 1$. Hence, v_2 has to be a sink so that the charge of $f = 1 - 3 = -2$. Let $v^{'}$ the neighbor of v_2 other than v_1 and v_3. As the same argument, $v^{'}$ has a C-forbidden neighbor $u^{'}$ as well. Notice that each u and $u^{'}$ will send one unit to f either by rule (D) or (F) depending on their degree such that the final charge of $f = -2 + 2 = 0$ is non-negative, which is a contradiction.

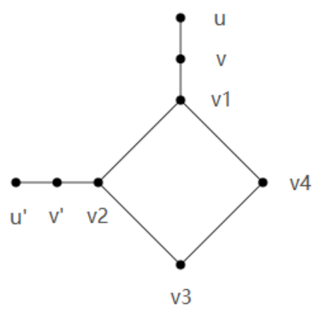

Figure 37: not C-secure tetragram

Thus, we've shown that every C-forbidden neighbor u of v is on C.

Observation 7. *Each 4-face f that shares an edge with C has final charge at most -2t, where $t \in \{1, 2\}$ is the number of sinks of f.*

At least one of C-forbidden neighbor u of v is adjacent to neither v_2 nor v_4 inasmuch as (v_1, v_2, v_3, v_4) is not a C-secure tetragram. Let C, C_1, C_2 be three cycles composed of C and the path v_4v_1vu.

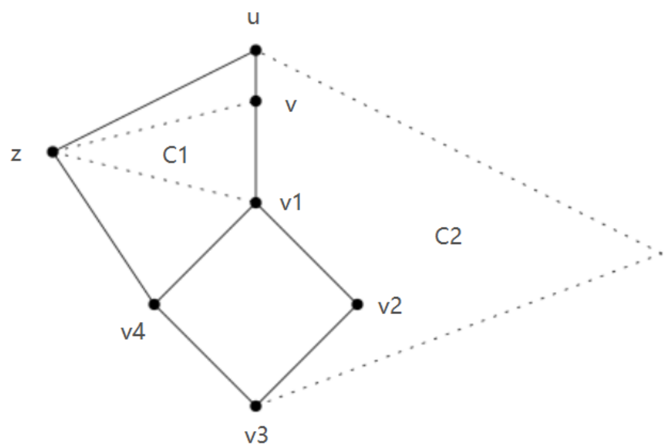

Figure 38: C, C_1, C_2

Observe from the **Figure 38** that v_2 is contained in C_2. So from **Corollary 7.1.1.** that the length of C_2 is at most six. Assume that the length of C_2 is exactly six, it results that v_2 has degree three and is adjacent to u, which is a contradiction.

For this reason, the length of C_2 is at least seven. We conclude that the length of C_1 is at most 5, because the length of C is at most six: $\ell_{C_1} + 7 - 2 \cdot 3 \leq 6 \iff \ell_{C_1} \leq 5$. By means of the constraint of u, the length of C_1 is exactly five as in **Figure 38**. Thereby, there is exactly a vertex z that is adjacent to u, v_4 and has degree two. If z would have degree at least three, then it would form a triangle or against the planarity. Let γ be the tetragram and $f(\gamma)$ be the face that is bounded by C_1.

Definition 7.5. A tetragram γ (v_1, v_2, v_3, v_4) is *bad*, if $f(\gamma)$ is defined.

Observation 8. *Bad tetragrams are faces of G that have always negative final charge. [3]*

Assume that the number of bad tetragrams in G is b. The initial charge of face $f(\gamma) = 9 \cdot 5 - 36 = 9$. Since

* v_1 is a sink: $v_1 \notin V(C)$ and has degree three.
* $z \in V(C)$ and has degree two.
* $v \notin V(C)$ and has degree three.

$f(\gamma)$ sends each three units to u, v, v_1 using rule (A). In addition, $f(\gamma)$ receives one unit either from v_3 using rule, if v_2 has degree exactly three; or from v_2 using rule (F), if v_2 has degree at least four so that the final charge of $f(\gamma)$ is -1. What's more, if there is another tetragram $\gamma^{'}$ that is different than γ such that $f(\gamma) = f(\gamma^{'})$. As a result, the final charge of $f(\gamma)$ is at most $-b$.

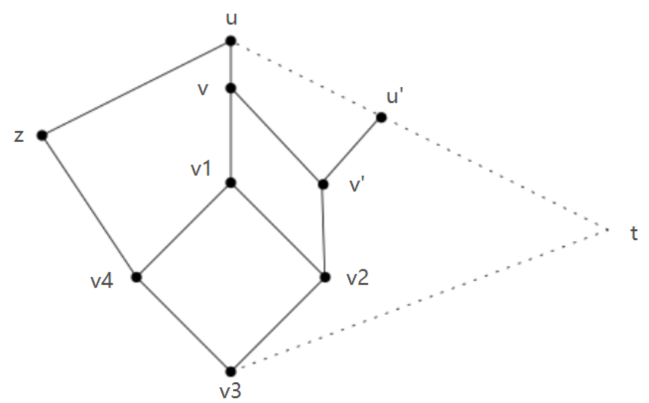

Figure 39: C-secure octagram $(v, v^{'}, v_2, v_1)$

Notice that v_3, v_4, u have degree at least three. From **Corollary 7.1.2.** informed that the total charge of G is then at most -3, which follows $b \geq 3$.

There must be another bad tetragram for $b > 1$. Hence, the final charge of G is at most -4. It leads $b \geq 4$. Let $u^{'}$ be the unique neighbor of u in $C \backslash z$. Notice that $v_3 v_4$ and $uu^{'}$ are the only edges of C incident to a bad tetragram. We conclude that G has a vertex $v^{'}$ of degree three with neighbors $v, v_2, u^{'}$ and hence $v^{'}$ is a sink and C-admissible. And it implies that according to the definition of sink, v_2 is then C-admissible. Notice that $u^{'}v^{'}v_2 v_3 t$ bounds also a face which is similar to the 5-face $zv_4 v_1 vu$. It follows that $(v, v^{'}, v_2, v_1)$ is a C-secure octagram, because $v, v^{'}, v_2, v_1$ are all C-admissible as proved. All in all, in any case, there will be a C-secure multigrams in G, as claimed.

□

References

[1] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*. Vol. 290. Macmillan London, 1976.

[2] Reinhard Diestel, Alexander Schrijver, and Paul Seymour. "Graph theory". In: *Oberwolfach Reports* 7.1 (2010), pp. 521–580.

[3] Zdenek Dvorak, Ken-ichi Kawarabayashi, and Robin Thomas. *Three-coloring triangle-free planar graphs in linear time*. 2013. arXiv: 1302.5121 [math.CO].

[4] Herbert Grötzsch. "Ein Dreifarbensatz für dreikreisfreie Netze auf der Kugel". In: *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe* 8 (1959), pp. 109–120.

[5] John Hopcroft and Robert Tarjan. "Efficient Planarity Testing". In: *J. ACM* 21.4 (Oct. 1974), pp. 549–568. ISSN: 0004-5411. DOI: 10.1145/321850.321852. URL: https://doi.org/10.1145/321850.321852.

[6] Lukasz Kowalik. "Fast 3-coloring Triangle-Free Planar Graphs". In: *Algorithmica (New York)* 58 (Nov. 2010), pp. 770–789. DOI: 10.1007/s00453-009-9295-2.

[7] Oscar Levin. *Discrete Mathematics: An Open Introduction*. CreateSpace, 2018. ISBN: 978-1792901690.

[8] *MIT 6.042J Mathematics for Computer Science, Fall 2010*. URL: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-042j-mathematics-for-computer-science-fall-2010/.

[9] Carsten Thomassen. "Grötzsch's 3-Color Theorem and Its Counterparts for the Torus and the Projective Plane". In: *J. Comb. Theory, Ser. B* 62 (1994), pp. 268–279.

[10] Carsten Thomassen. "On separating cycles in graphs". In: *Discrete Mathematics* 22.1 (1978), pp. 57–73. ISSN: 0012-365X. DOI: https://doi.org/10.1016/0012-365X(78)90047-X. URL: https://www.sciencedirect.com/science/article/pii/0012365X7890047X.

[11] Haoze Wu. *An Introduction to the Discharging Method*. 2020. arXiv: 1711.03004 [math.HO].

A Appendix

Following are illustrations of reductions of the multigram. Noted that in the following illustrations, we just remove vertices with degree at most one before executing the algorithm instead of vertices with degree at most two, because we want to show how vertices in multigrams are identified.

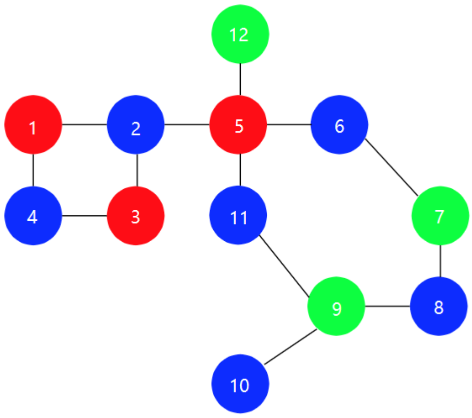

Figure 40: The input graph

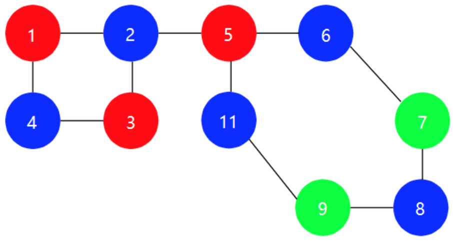

Figure 41: Remove vertices with degree at most one

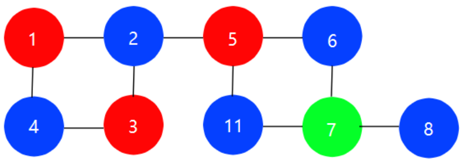

Figure 42: Identify vertices 7 and 9 which are in a safe hexagram

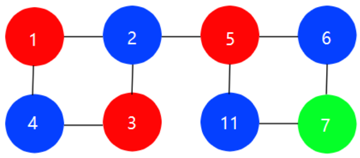

Figure 43: Remove vertices with degree at most one

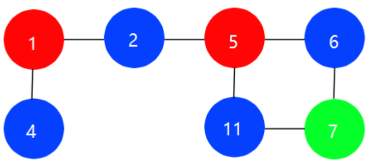

Figure 44: Identify vertices 1 and 3 which are in a safe tetragram

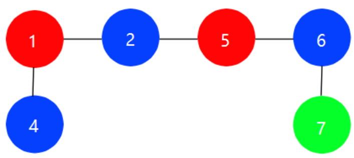

Figure 45: Identify vertices 6 and 11 which are in a safe tetragram