Tampere University

Mina Shahmoradi

PLANARITY TESTING OF A GRAPH

Bachelor thesis
Faculty of Engineering and Natural Sciences
Supervisor: University teacher Petteri Laakkonen
April 2021

ABSTRACT

Mina Shahmoradi: Planarity testing of a graph
Bachelor thesis
Tampere University
International bachelor programme of Science and Engineering
April 2021

Graphs provide a way to model connections between objects in different systems, e.g., roads, power grids, and networks. Mathematically graphs are structures that present a set of elements and their connections. Typically, graphs are illustrated graphically by denoting the elements by dots and the connections between them by lines. Graph theory and the properties associated to graphs will give a new approach so that it can be used as tool to reformulate different problems.

This thesis focuses on the results that can be used to test if a given graph is planar or not. A graph is planar if we can give it a graphical illustration in a plane so that the lines connecting the elements do not intersect. The thesis presents some fundamental properties of the planar graphs. A particularly important result is the Kuratowski's theorem, which gives a necessary and sufficient condition for a graph to be planar in terms of certain minor graphs called Kuratowski's minors.

A planarity testing algorithm is presented as a major result of this thesis. Its functionality is illustrated by an example that explains in detail how the algorithm constructs a representation of the graph in a plane with no intersecting lines or returns a Kuratowski's minor if the graph is not planar. then, some properties of planar graphs were reviewed, for example number of edges in a planar graph which is bounded to the number of vertices. And finally, after defining coloring of a graph, coloring property of planar graphs were review, such as them being 5 and 4-colorable.

Keywords: graph theory, planar graphs, graph coloring, planarity, colorability

PREFACE

The idea of this thesis came from my desire to learn and explore mathematical backgrounds of the well-known methods and algorithms which I knew from their applications. This thesis was not possible without the support of my supervisor and teacher Petteri Laakkonen who shown compassion and understanding despite all challenges though researching and writing of this thesis. Also, I could not succeed without the support of my parents, who are far from me in these days of pandemic, but their love is close in my heart. I can not thank Mo enough for his support and understanding.

Tampere, 25th April 2021

Mina Shahmoradi

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

$\mathbf{E(G)}$	Set of edges of graph G
$\mathbf{G}$	Graph
$\mathbf{V(G)}$	Set of vertices of graph G
e	Edge of a graph
m	Number of edges in a graph
n	Number of vertices in a graph, graph's order
r	Number of regions made by a graph
v	Vertex of a graph
TAU	Tampere University
TUNI	Tampere Universities
URL	Uniform Resource Locator

1. INTRODUCTION

The idea of graph theory was firstly presented when Euler tried to solve the problem of 7 bridges of Königsberg (Euler, 1735). He wanted to prove if it was possible so that all bridges in the Königsberg city can be passed once and only once by going through the network, which he ended up proving it was not possible. The formulated idea of this problem as a graph is presented in Figure 1.1 where each vertex dot is representing a city part and each line is a bridge connecting those parts.

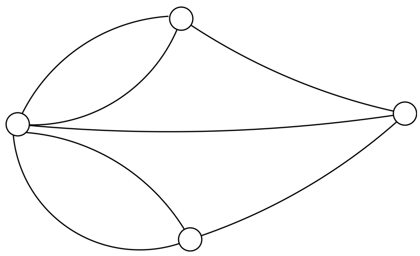

Figure 1.1. *Graph presentation of 7 bridges of Königsberg*

Although graph theory was initially used to represent a simple network, nowadays it has been used to represent more complicated technological networks such as power grids[1], road networks,[2] and 3D data networks [3]. Apart from presenting different networks, graph theory has been used as a new way of data representation. This will give the opportunity to deal with the data using the new formulation. Using properties of graph theory gives the opportunity to have a new perspective and approach to solve problems. Using adjacency matrix to present graphs and solving problems by utilizing their relations in vector space [4] and using graph representation in data structure [5] are a widely used examples of this approach.

One of the specific properties of graphs that comes with their representation as nodes (dots) and vertices (lines connecting the dots) is whether they are planar or not. A graph is known to be planar if it can be presented on a plane where its vertices and edges are distinctly drawn and the edges are not crossing each other. Many authors have developed different algorithms to test planarity of a graph, these algorithms differ based on their time complexity to run, methods of implementation, and their proves' complexity [6].

Planar graphs have their own specific properties dealing with other concepts in graph theory such as being colorable, having cubic duality, and having bridges of cycles [7,

p. 243–286]. These features have been used in different areas such as optimization of robotic algorithm [8], identification of coherent structures from sparse data [9], and even finding genetic patterns[10].

This study gives an introduction so some preliminary definitions of graphs and uses these properties to overview an algorithm for planarity testing of a graph. The reviewed algorithm runs in polynomial time complexity and deals with a given graph as input. It shrinks the graph to a reasonable size so that it has a planar embedding and by adding the removed vertices back, a graph will be returned as an output. If the input graph is planar, output will be planar embedding of it, although if the given graph is not planar, a minor of it will be returned, showing that it was not a planar graph.

In addition to the planarity testing algorithm, some properties of planar graphs will be explored. It will be shown that the maximum edges of a graph with specific number of vertices is limited to a certain amount. Furthermore, it will be shown that planar graphs are colorable with certain number of colors. These properties of planar graphs are supported using proofs and examples.

The structure of the remaining text is as follows. First the fundamental concepts that are required in latter parts are introduced in Chapter 2. The main results of this thesis are given in Chapter 3. First, we introduce planar graphs. In Section 3.1 we explore Kuratowski's Theorem and then an algorithm to test planarity of a given graph graph. Then in Section 3.1 we introduce some properties for planar graphs such as Euler's Formula and its direct result and how they can be used to check non planarity of a given graph. Then in section 3.3 coloring of a graph is introduced and then coloring of planar graphs is evaluated in more detail. Finally, the main results of the thesis are summarized in Chapter 4.

2. FUNDAMENTAL CONCEPTS

One of the most convenient models to represent a graph is to use points and lines connecting those points, or as we call them here, vertices and edges. Vertices can be connected to each other or to themselves by edges. Graph G is a structure made of vertices and edges showing their relations. Graphs can be explicitly presented as an ordered set of $(V(G), E(G))$ where $V(G)$ is a set of its vertices and $E(G)$ is a set of its edges.

Then different connection between vertices through edges can be define by the incidence function. Although edges can be directional, in this study we consider them both ways, meaning whenever two vertices of v_1 and v_2 are joined by an edge e, v_1 is connected to v_2 and v_2 is connected to v_1. More importantly graphs can be depicted graphically, leading to a better understanding of their properties and their result.

Example 1. *The given examples in Figure 2.1 are graph G_1 and G_2 representation. Graph G_1 is K_4 and graph G_2 is a tree.*

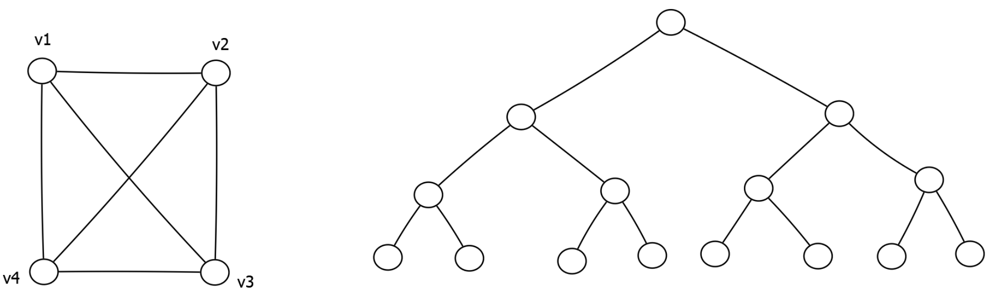

Figure 2.1. *Graph presentation of G_1 and G_2*

In the following some properties of graphs and their relations are defined which we will use them in later chapters.

In naming an edge of a graph usually start and end vertex's name will be used, e.g. the edge connecting v_1 and v_2 is denoted by $e_{1,2}$. Then if an edge has identical ends, it is called a loop, whereas if it has distinct ends it will be called a link. Two or more number of links sharing same pairs of ends are called parallel edges.

Example 2. *In the represented graph in Figure 2.2 e_1 is a loop, $e_{1,2}$ and $e1_{2,3}$ are links, and $e1_{2,3}$ and $e2_{2,3}$ are parallel edges.*

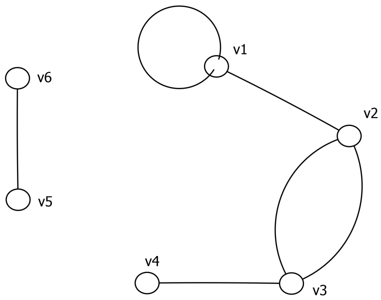

Figure 2.2. *A disconnected graph, having a loop and parallel edges*

Definition 3 (Simple graph)**.** *A graph is called* simple *if it has no loops or parallel edges [7, p. 3].*

Definition 4 (Contraction)**.** *Edge contraction is an operation in which an edge will be removed and simultaneously the two end vertices to that edge will be merged. The connected edges to each of those vertices will remain connected to the resulting merged vertex. If we contract the edge e from the graph G, then the result will be G/e.*

Definition 5 (Minor)**.** *Whenever graph H can be formed by contracting edges or deleting vertices in a given graph G, H is called minor of the graph G [7, p. 268].*

Example 6. *The given graph in Figure 2.3 is called Petersen graph. If we contract spoke edges between the pentagon and the star in the center which are $e_{1,6}, e_{2,7}, e_{3,8}, e_{4,9}, e_{5,10},$ from Petersen graph, we will form resulting graph in Figure 2.4 showing a minor of it, known as K_5.*

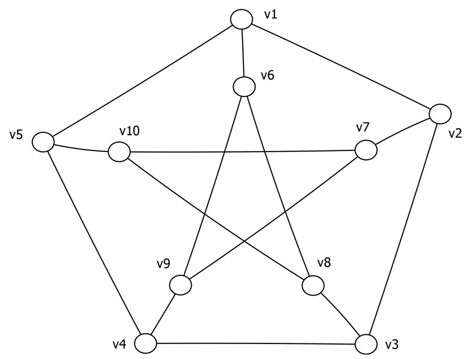

Figure 2.3. *Petersen graph*

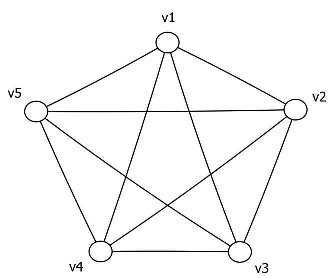

Figure 2.4. *K_5 Graph*

Definition 7 (Path)**.** *Path is a sequence of distinct vertices of a graph so that consecutive vertices are adjacent in the graph.*

Definition 8 (Cycle)**.** *Cycle is a path of 3 or more vertices in which starting and end vertex are the same [7, p. 4].*

Length of a path or a cycle is the number of distinct vertices in them. Also, a 3-cycle is called triangle. As a side note, if a graph does not contain a triangle, it is called triangle-free.

Example 9. *The sequence of v_1, v_2 and v_3 is an example for a path in K_5 in Figure 2.4. While the sequence of v_1, v_2, v_3, v_1 is a triangle and v_1, v_2, v_3, v_5, v_1 is a 4-cycle in K_5.*

Definition 10 (Connected graph)**.** *If for any division of vertices of graph G into two non-empty subsets of X and Y so that they have no common elements and the union of X and Y is $V(G)$, there is an edge with one end in X and the other in Y then G is connected. If a graph is not connected, it is disconnected [7, p. 5].*

Example 11. *Depicted graphs in Figures 2.3 is a connected graph, whereas the given graph in Figure 2.2 is an example of a disconnected graph, in which v_5 and v_6 are disconnected from the rest of the graph.*

Definition 12 (K-connected graph)**.** *Considering two distinct vertices x and y in graph G and the paths connecting them P and Q, if these paths have no internal vertices in common, meaning $V(P) \cap V(Q) = \{x, y\}$, then P and Q are locally disjoint. Now, we define $p(x, y)$ as local connectivity between x and y to be the maximum number of pairwise internally disjoint xy-paths in G. The graph G is k-connected if $p(x, y) \geq k$ for any two distinct vertices x and y.*

Definition 13 (Vertex cut)**.** *A vertex cut is a subset of vertices of a graph which, if together with any incident edges removed from the graph, it will disconnect it.*

Definition 14 (Sub-graph)**.** *A graph H resulting from deleting an element from a given graph G will be a sub-graph of G [7, p. 40]. Now, if G has more edges than H, then H is a proper sub-graph of G.*

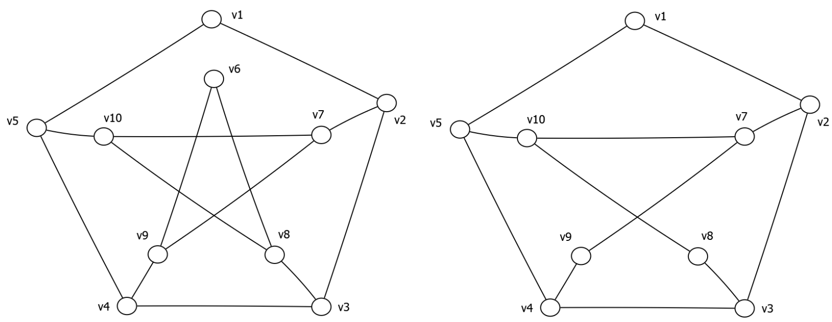

Figure 2.5. Graph G, $G/e1$, $G/v3$

Example 15. *In the given graph in Figure 2.5 two different sub-graphs of Petersen graph are shown. The sub-graph on the left is a result of deleting $e_{1,6}$, which can be denoted as $G\backslash e_{1,6}$. Then the sub-graph on the right is the result of vertex deletion of v_6 from the Petersen graph, which can be denoted as $G - v_6$.*

Definition 16 (Bridge)**.** *Consider connected graph G to have a proper sub-graph H, by deleting $E(H)$ from $E(G)$ we will be resulted to these classes,*

- *for each component F of $G - V(H)$, there is a class of edges of F together with the edges linking F to H*
- *any remaining single edge which its both ends belong to $V(H)$*

These resulting sub-graphs of G formed by these classes are the bridges of H in G.

If we define segment of a k-bridge B with $k \leq 2$, to affect a partition of the cycle C into k edge-disjoint paths, we can observe bridge relations to one another as follow. A bridge with k attached vertices is a k-bridge. Also, two bridges with same attached vertices are defined as equivalent bridges. If all vertices attached to a bridge lie in a single segment of another bridge, these two bridges avoid each other. If two bridges do not avoid, they overlap. Finally, as bridge B_1 have u_1 and v_1 as its attached vertices while u_2 and v_2 are attached to B_2; if these are distinct vertices occurring in a cyclic order in C such as u_1, u_2, v_1, v_2, then B_1 and B_2 are skew [7, p. 263-264].

Example 17. *In the Figure 2.6 a graph is shown, having cycle C while several bridges are attached to it. Bridges B_1 and B_3 avoid each other, whereas B_3 and B_4 overlap. Also, B_3 and B_4 are 2-equivalent and B_2 and B_1 are skew.*

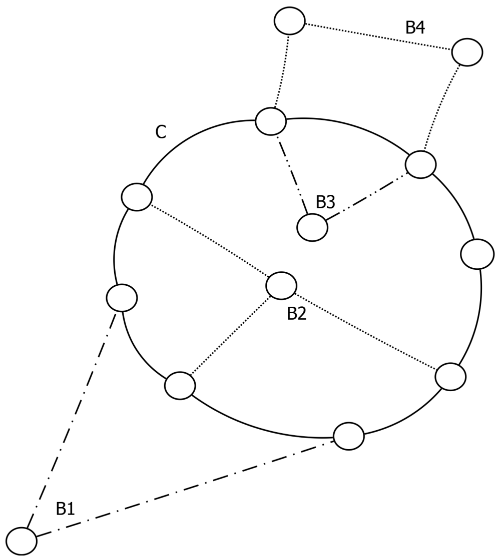

Figure 2.6. *Bridges of a graph*

Definition 18 (Isomorphism)**.** *An isomorphism of two graphs $f : V(G) \to V(H)$ is a bijection between the vertex sets of them such that any two vertices like u and v of G*

are adjacent if and only if $f(u)$ and $f(v)$ are adjacent in H.

By defining a plane simple closed curve such that it is a continuous non-self-intersecting loop in the plane, known as a Jordan curve, we can review Jordan Curve Theorem.

Theorem 19 (Jordan Curve theorem)**.** *Every Jordan curve divides the plane into two regions, an interior region bounded by the curve and an exterior region containing all the exterior points. This theorem states that every continuous path connecting a point of one region to a point of the other, eventually will intersect the curve somewhere [7, p. 245].*

3. PLANARITY OF A GRAPH

After defining some useful concepts, we can explore planarity of a graph and how to test if a graph is planar or not. We can start by defining planar graphs mathematically.

Definition 20 (Planar Graphs)**.** *If a graph can be drawn on a plane, still having these two conditions it will be considered a planar graph.*

1. *Vertices can be displayed as distinguishable points displayed on a plane.*
2. *Edges do not intersect with each other unless on their endpoints.*

Example 21. *Both graphs shown in Example 1 are planar graphs. Although there are some crossing edges in the particular drawing of the graph G_1 in Figure 2.1 which can be represented in its planar drawing, known as planar embedding, in Figure 3.1.*

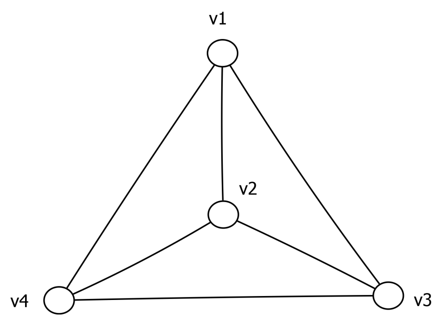

Figure 3.1. *Planar presentation of graph K_4*

On the other hand, there are some graphs that no matter how we draw them in a plane, some of their edges will cross.

Example 22. *The K_5 graph shown in Figure 2.4 is not a planar graph. Considering K_4 as shown in Figure 3.1, we need to add a vertex to it and connect the new vertex to all existing vertices to make K_5. We can place the new vertex in any region formed in a planar embedding of K_4. As it can be observed from the Figure 3.1 there are 4 distinct regions in K_4. According to Jordan Curve Theorem, resulting edges to connect the fifth vertex will cross an edge, no matter in which region we put the fifth vertex, which shows that K_5 is not a planar graph.*

Graph $K_{3,3}$ is another example of non-planar graph, shown in Figure 3.2. Non planarity of $K_{3,3}$ will be shown later in Example 32. K_5 and $K_{3,3}$ are often referred as Kuratowski's *minors.*

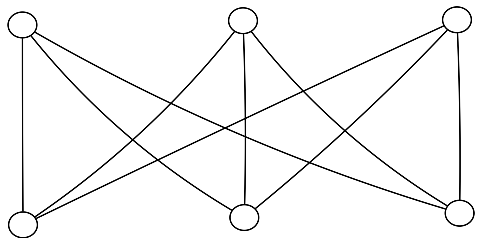

Figure 3.2. *Graph* $K_{3,3}$

3.1 Planarity Testing Algorithm

Planarity testing is a method to check if a graph is planar or not. This subject has been studied by many authors, while one of the earlier theorems were conducted by K. Kuratowski [11].

Theorem 23 (Kuratowski's theorem)**.** *A graph is planar if and only if it does not have a minor of* K_5 *or* $K_{3,3}$*. [7, p. 268]*

The proof to Kuratowski's Theorem is out of the scope of this study, although it can be formulated to some smaller pieces for a better understanding as follow. Firstly, a minimal non-planar graph can be defined as a non-planar graph in which every nontrivial sub-graph is planar. Then, it can be proved that each minimal non-planar graph is 3-connected. And finally Kuratowski's theorem can be proved by showing that every 3-connected graphs with no Kuratowski minor is a planar graph.[11, p. 246–248]

Many algorithms have been presented to check planarity of a graph. Mostly these algorithms are based on Kuratowski's theorem, although they have developed into other forms and their implementation might differ hugely. Time complexity of these algorithm differ massively, which will affect their practicality to implement and run.

First algorithms were based on Kuratowski's characterization of planar graphs which resulted to an algorithm with an exponential compilation time [6]. Since then, there has been significant improvements developing algorithms to test planarity of a graph. Although the other issues of planarity testing algorithms are their complexity to prove, develop, and implement.

Even though the following algorithm presented here is based on the redult of Kuratowski's theorem, it is not complex to prove that the algorithm works as intended. In fact the algorithm can be prove using Lemma 25. In the worst case scenario, its main parts such as contraction and expansion phase would compile in polynomial time so the time complexity of the algorithm is polynomial time.

Lemma 24. *If G_1 and G_2 are planar graphs with intersection isomorphic to K_2, then the union of G_1 and G_2 is planar.*

Proof. Since the intersection of G_1 and G_2 is isomorphic to K_2 it consists of one edge e and its endpoints. Consider a planar embedding of G_1. It has a region r that has the edge e on its boundary. Since e is the only edge and its endpoints are the only vertices that are common in G_1 and G_2, the planar embedding of G_2 can be drawn in r resulting into a planar embedding of the union. □

Lemma 25. *Let G be a graph with a 2-vertex cut of {x, y}. Then if and only if all of its marked 2-vertex components of G are planar, G itself is planar [7, p. 270].*

Proof. First, we assume that G is planar. Define H as a {x, y}-component with marker edge e of graph G. Then in another {x, y}-component let $P_{x,y}$ be a path. Although union of P and H is a sub-graph of G, it is isomorphic to a subdivision of $G + e$, so $G + e$ is isomorphic to a minor of graph G. This shows that all marked 2-vertex component of graph G are isomorphic to a minor of graph G. As all minors of a planar graph are planar themselves, we can conclude that all 2-vertex components of G are planar.

Now assuming that all k marked 2-vertex components of graph G are planar. Define e as a common marker edge. Using Lemma 24 we conclude that union of the first two components will be planar as well. By using induction on all k element of graph G, $G + e$ will be planar. This means that graph G is planar itself. □

Lemma 25 implies that a graph is planar if and only if all of its 3-connected subgraphs are planar. This means that the planarity of a graph can be verified by considering each of the 3-connected component of the graph individually. As a result, the algorithm to check whether a graph is planar or not can be presented simply as follow; first the input graph is reduced to a four vertices graph by contracting edges one by one while maintaining the graph 3-connected. This phase, labelled as contraction phase will be done in polynomial time. Then contracted edges are then expanded in reverse order one by one.

In expansion phase two outcomes are possible when expanding an edge: planarity of the graph might preserve or not. If planarity of the graph is preserved then the algorithm proceeds to the next contracted edge, if planarity is not preserved, expansion will result to a Kuratowski's minor where two overlapping bridges are found. If the first outcome emerges for all edges in the graph G, then the output of the algorithm will be a planar presentation of the graph G, whereas the second possibility yield to a non-planar minor, resulting the whole graph to be non-planar by Theorem 23.

Algorithm 26 (Planarity recognition algorithm of a graph)**.** *The input to this algorithm is a 3-connected graph G with 4 or more vertices. The output of this algorithm will b either a* *planar representation of graph G or a Kuratowski's minor with polynomial time complexity [7, p. 270].*

Initialization phase:

$i = 0$ *and* $G_0 = G$

Contraction phase:

while $i < n - 4$ *do*

    In G_i *find the link* $e_i := x_i y_i$ *so that* G_i / e_i *is a 3-connected graph*

    set $G_i := G_i / e_i$

    $i \leftarrow i + 1$

end while

Expansion phase:

Find a planar representation $\tilde{G}_{n-4}$ *of the 4-vertex graph* G_{n-4}

$i \leftarrow n - 4$

while $i > 0$ *do*

    define z_i *to be the resulting vertex of* $\tilde{G}_i$ *from contraction of the edge* e_{i-1} *of* G_{i-1}, *let* C_i *be the bounded cycle of* $\tilde{G}_i - z_i$ *that include all neighbours of* z_i *in* $\tilde{G}_i$

    Let B_i *and* B_i' *represent the bridges of* C_i *respectively containing the vertices* x_{i-1} *and* y_{i-1} *in the obtained graph from deleting* e_{i-1} *of* G_{i-1} *and all the other edges connecting* x_{i-1} *and* y_{i-1}

    if B_i *and* B_i' *are skew then*

        Find a $K_{3,3}$*-minor* K *of* G_{i-1}

        Return K

    end if

    else if B_i *and* B_i' *are a 3-bridges equivalent then*

        find K *such as it is* K_5*-minor of* G_{i-1}

        return K

    end if

    else

        expand $\tilde{G}_i$ *to* $\tilde{G}_{i-1}$

        $i \leftarrow i - 1$

    end else

end while

return $\tilde{G}_0$

Example 27. *Using presented Algorithm 26, we will check planarity of the given graphs in Figure 3.3 and we will either find the planar embedding of them or their Kuratowski's minor.*

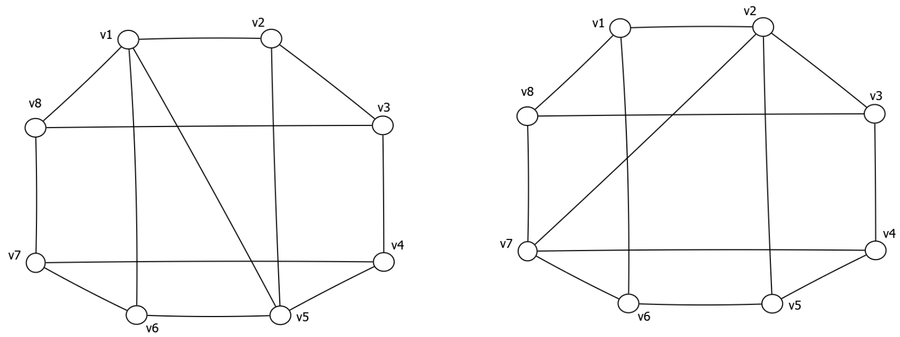

Figure 3.3. *Planar presentation of graph G and H*

First we will check the graph G in Figure 3.3. We initialize $i = 0$ and $G_0 = G$. Then while i is less than $n - 4 = 4$ we go through lines $2 - 5$ of the algorithm and repeat the contraction phase. Although we can choose any edges for this part of algorithm as long as the resulted graph stays 3-connected, as an example here we start contraction by eliminating $e_{5,6}$ and combining v_5 and v_6 to a single vertex, naming it v_5. We update $i = 1$ and save G_1 as the result of contraction, which can be seen in Figure 3.4.

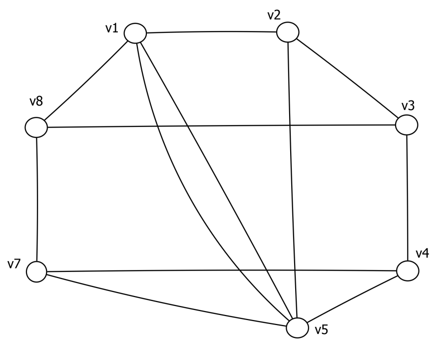

Figure 3.4. *G_1, result of the first contraction phase on graph G*

By repeating contraction phase till $i = n - 4 = 4$, we need to do the contraction phase 3 more time. First we contract $e_{3,4}$, and name the combined vertices v_3, resulting our graph to be updated as G_2 and $i = 2$. Then we contract $e_{7,8}$ and let the combined vertices named v_7, having the resulted graph as G_3 and update i to be 3.

Then as the last turn of contraction phase, we will contract $e_{1,2}$ and let the combined vertex to be v_1, update our graph to G_4 and let $i = 4$. At this point, as i has reached the while limit, while loop ends. The resulting graph of G_4 is shown in Figure 3.5.

Continuing the algorithm and starting the expansion phase, in line 7 for the given graph G_4 we will find a planar embedding of $\tilde{G}_4$ as shown in Figure 3.6.

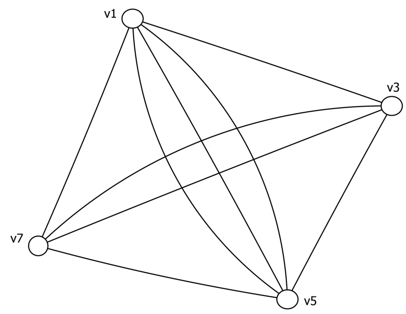

Figure 3.5. *G_4, result of the last contraction phase on graph G*

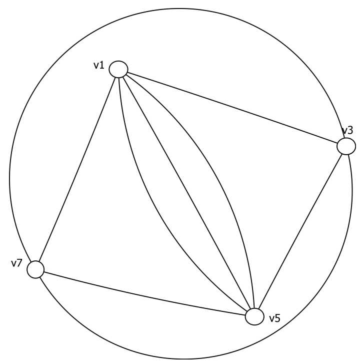

Figure 3.6. *$\tilde{G}_4$, the planar representation of graph G_4*

Now as $i = n - 4 = 4$ we go through the while loop in line 9 and repeat the expansion phase till $i = 0$, going in the reverse order over contracted edges in previous phase. As the last contracted edge is $e_{1,2}$ and v_2 was eliminated, we form z_4 to be v_1. Now, we define bounded cycle of C_4 in $\tilde{G}_4$ such that it contains all vertices connected to z_4 in graph $G_4 - z_4$. This will result C_4 to be v_3, v_5, v_7, v_3. Next, we define B_4 in $G_4 - z_4$ such that it is attached to all the vertices connecting v_1 to C_4; so here v_3, v_5 and v_7 will be attached to B_4; whereas v_3 and v_5 are attached to B'_4. As B_4 and B'_4 neither are skew, nor 3-bridge equivalent to each other, we expand $\tilde{G}_4$ to $\tilde{G}_3$ which is shown in Figure 3.7.

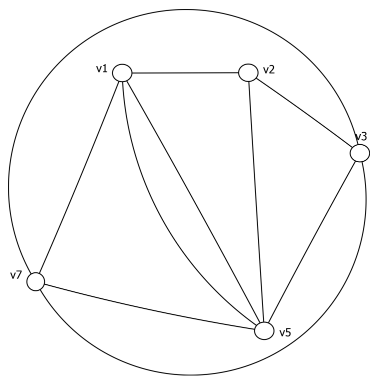

Figure 3.7. *$\tilde{G}_3$, result of the first expansion phase on $\tilde{G}_4$*

Repeating the while loop in line 9 and expanding previously contracted edges, finally we will have the planar embedding of the graph G as shown in Figure 3.8 at the end of the algorithm.

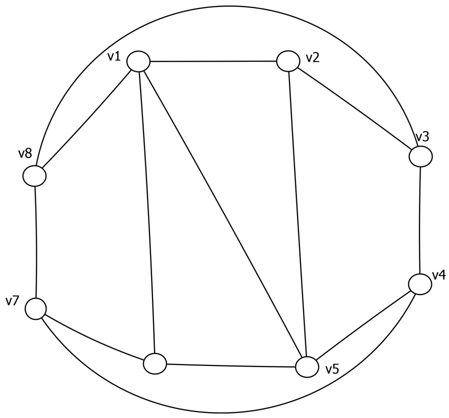

Figure 3.8. *Planar embedding of graph G*

Now we check graph H from Figure 3.3 and go through the Algorithm 26 to either find its planar embedding or its Kuratowski's minor.

Repeating the same process, we start by initializing $H_0 = H$ and $i = 0$. Then we go through contraction phase, we will contract $e_{5,6}$, letting the combined vertices be named v_5. Then similarly we will contract $e_{3,4}$, $e_{7,8}$ and $e_{1,2}$. The 4-vertex graph result of the contraction phase called H_4 has been shown in Figure 3.9.

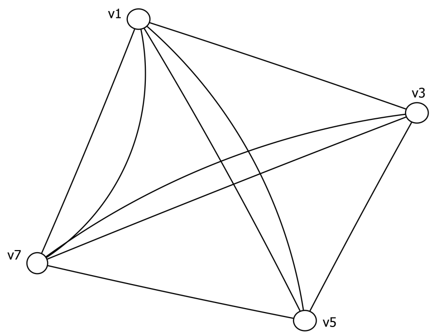

Figure 3.9. *Graph H_4, result of the contraction phase on graph H*

Then in expansion phase we will find the planar representation of H_4, named $\tilde{H}_4$. After that, we start to expand contracted edges and their attached vertices in reverse order, one by one. After adding v_2, the resulted graph will be $\tilde{H}_2$ shown in Figure 3.10.

Then in next expansion phase we will try to add v_7 to $\tilde{H}_3$. In this stage $z_3 = v_7$ as C_3 is going to be a bounded cycle in $\tilde{H}_3 - z_3$, containing the neighbouring vertices to z_3 in G_3. So C_3 will be v_1, v_2, v_3, v_5, v_1. Here $v1$, v_2, v_3 and v_5 are attached to B_3, whereas v_1 and v_3 are attached to B_3'. As these vertices appear in cyclic order in C_3, B_3 and B_3' are skew. The resulting graph to add v_7 to $\tilde{H}_3$ will be $\tilde{H}_2$ shown below in Figure 3.11.

Rearranging the vertices of $\tilde{H}_2$ in Figure 3.11 we will find the resulting graph in Figure 3.12, in which by rearranging v_2 and v_7 we will find the $K_{3,3}$ minor of $\tilde{H}_2$. This result clearly shows that graph H is not a planar graph.

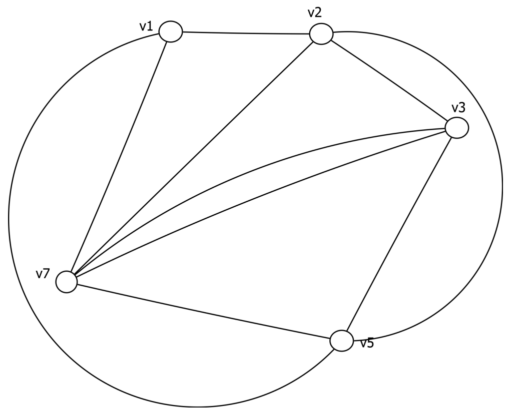

Figure 3.10. *$\tilde{H}_3$, result of the first expansion phase on $\tilde{H}_4$*

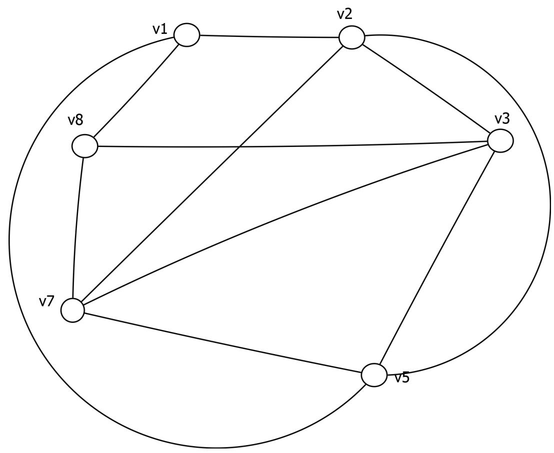

Figure 3.11. *$\tilde{H}_2$, result of the second expansion phase on $\tilde{H}_4$*

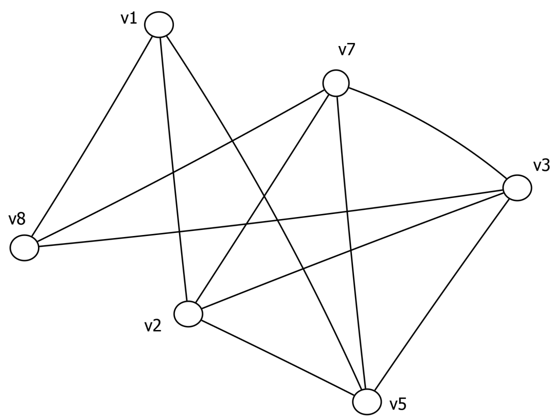

Figure 3.12. *Rearranged drawing of $\tilde{H}_2$*

3.2 Properties of planar graphs

Although Euler's Formula was introduced as a property for planar graphs, using it can yield to an efficient method for checking if a given graph is not planar. Though, Euler's Formula will not help to find the planar embedding for a given graph, also it can be inconclusive in case the inequality holds.

Theorem 28 (Euler's Formula)**.** *[7, p. 259] If G is a planar embedding of a graph where no edges cross each other, considering n as the number of vertices and m as the number of edges in G, then the graph divides the plane into r regions where:*

$r = m - n + 2$

Proof. Euler's Formula can be proved by induction on the number of edges.

Assuming that a connected planar graph has n vertices, the least number of edges that it can have is $m = n - 1$. This case happens when G is a tree, which does not have any cycle, resulting the plane to be just one region, so

$r = m - n + 2 = n - 1 - n + 2 = 1$

The formula holds for n vertices graph with $m = n - 1$ edges. Now for the induction step, we need to show that Euler's Formula holds for graphs with more edges, so we have $m > n - 1$. In this case G has to have a cycle. By removing an edge from the cycle of G, we form the connected graph G' with $r - 1$ regions, $m - 1$ edges, and n vertices. By the induction hypothesis the number of regions in G' is

$r - 1 = (m - 1) - n + 2$

By adding adding the removed edge to the graph we will close the cycle, so the number of regions and edges are increased by one. By adding one to both sides of the above equality we will get the Euler's formula. This completes the induction step and we have shown that the claim holds for all planar connected planar graphs with n vertices. □

Lemma 29. *Assuming that G is a planar embedding of a simple connected graph where no edges cross each other, considering $n \geq 3$ as the number of vertices and m as the number of edges in G, we will have:*

$m \leq 3n - 6$

Proof. Letting r to be the number of regions in the planar embedding of graph G, we define f_i to be the number of surrounding edges around the region i. If a region happens to be on both sides of an edge, then the edge is counted twice, so:

$\sum_{i=1}^{r} f_i = 2m$

We know that graph G is a simple graph and has more than two vertices so each region is bounded at least by 3 edges, so $f_i \geq 3$, resulting to $3r \leq 2m$. We already knew that $r = m - n + 2$, so by Euler's formula $3r = 3m - 3n + 6$. Combining it with the previous inequality we have $3m - 3n + 6 \leq 2m$, so we conclude that $m \leq 3n - 6$. □

Example 30. *Using Lemma 29 we can simply check if K_5 is planar or not. As we know K_5 has 10 edges and 5 vertices, so by checking $3n - 6$, we get 9 which is less than number of edges in K_5, from which we can conclude that K_5 is not a planar graph.*

Lemma 31. *If the graph G described in Lemma 29 is triangle-free, then $m \leq 2n - 4$*

Proof. Again, we define r to be the number of regions in the planar embedding of graph G, and f_i to be the number of surrounding edges around the region i. As G is triangle-free, then each region made on a plane by G is surrounded by at least four edges. In this case we have $2m = \sum f_i \geq 4r$, so by the Euler's formula we will have $m \leq 2n - 4$. □

Example 32. *As it was shown in Figure 3.2, $K_{3,3}$ has 6 vertices and 9 edges. If $K_{3,3}$ is a planar graph, by Euler formula 28 it should have $r = 5$ regions. As $K_{3,3}$ is triangle-free, then each region should be bounded to at least 4 edges, which means $4r \leq 2m$. But here we have $20 \leq 18$ which is a contradiction, showing our first assumption was incorrect and $K_{3,3}$ is not a planar graph.*

3.3 Coloring of a planar graph

Graph coloring name comes from its application in map coloring, where a problem of whether a map is colorable with only 4 colors so that no region next to each other have the same color or not was firstly mentioned by a student of A. De Morgan in 1852 [7, p. 287]. This question drew attention of different mathematicians to the problem and led to advancement of the idea. In coloring problems as label's values are not important, we may refer to them in any way we want. Here we choose to label the colors using numbers for simplicity.

Even though coloring of a graph can be represented as coloring of its vertices or coloring of its edges, here whenever we mention graph coloring, we are referring to coloring of vertices of a graph.

Definition 33 (Coloring)**.** *Coloring of a graph is a function assigning different colors to its vertices. Also, if adjacent vertices in a graph have different colors, then the coloring of the graph is* proper*.*

Definition 34 (K-Coloring)**.** *If a graph has a* proper K-coloring*, then it is considered K-colorable [11, p. 191]. We use notation $V(G) \to S$ if the graph is k-colorable with colors in the set S such that $|S| = k$.*

Note that in a proper coloring, every color class is an independent set, so that every two adjacent vertices belong to a different color class. This means that G is k-colorable if and only if $V(G)$ is union of k independent sets.

Example 35. *The following representation in Figure 3.13 of C_5 and Petersen graphs are 4-coloring examples of them.*

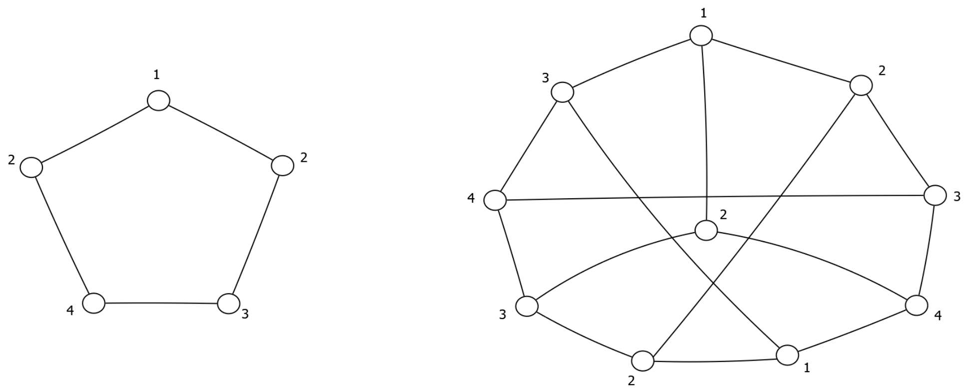

Figure 3.13. *4-coloring presentation of C_5 and Petersen graphs*

An important note is that graphs with loops are not colorable as a vertex cannot have different colors than itself. So, all colorable graphs are without any loops. This means all graphs we are mentioning in this chapter are considered to be loop-less. Multiple edges between two specific vertices do not affect the graph coloring. Therefore, it is typical that the graph is assumed to be simple when considering coloring of it. Also, most of the statement made for a simple graph's coloring will remain valid if it has multiple edges as well.

Theorem 36 (K-Chromatic)**.** *The least possible value k such that a graph is K-colorable is the chromatic number of the graph [11, p. 191].*

On a side note, a *bipartite graph* is a graph which can be divided into two subsets of vertices where each end of those edges connecting vertices belong to just one of the sets. It follows that a graph is bipartite if and only if it is 2-colorable.

Example 37. *As C_5 and Petersen graphs are not bipartite, their chromatic number is at least three. While they can be represented as 3-colorable graphs as follow in Figure 3.14, their chromatic number is exactly 3.*

Theorem 38 (Five Color Theorem)**.** *Every planar graph is 5-colorable [11, p. 257].*

Proof. Using induction on n we can proof that a planar graph is 5-colorable.

Basic step is whenever $n(G) \leq 5$, such graphs will be always 5-colorable.

In the induction step we assume that G is a planar graph with $n(G) > 5$ vertices as any planar graphs with fewer vertices are 5-colorable. As the graph is planar, Lemma

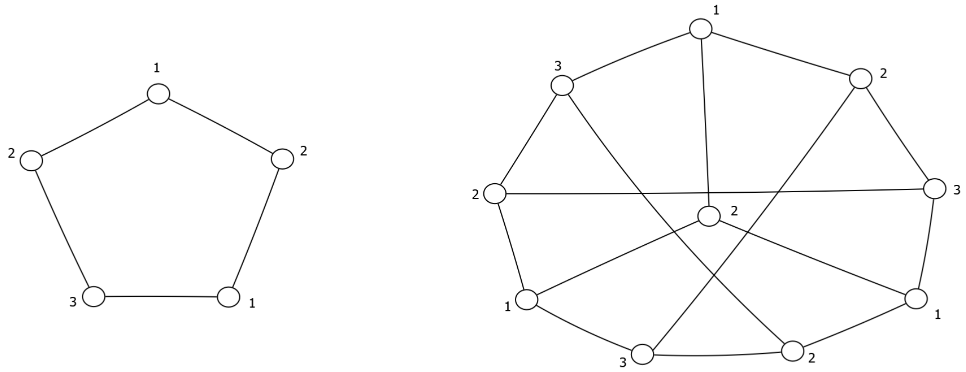

Figure 3.14. *3-coloring presentation of C_5 and Petersen graphs*

31 indicate that maximal degree of a vertex in G is 5. If the degree of v is less than 5, there is a free color to be assigned to v, so that no neighbour of v has the same color. Thus, G would be 5-colorable. So, we assume that the degree of v is five. The induction hypothesis implies that $G - v$ is 5-colorable.

Now let v_i for $i = 1, 2, ..., 5$ be neighbouring vertices to v in clockwise order. Let's define function f such that it is a proper coloring for $G - v$, colors can be assigned using f to G by $f(v_i) = i$. Assuming that f cannot be extended to a proper 5-coloring of G, f will assign each color to a neighbouring vertex of v, so that $f(v_i) = i$.

Now define $G_{i,j}$ to be the maximal connected sub-graph of $G - v$ containing vertex v_i that is induced by vertices having i and j colors. Switching colors for any two vertices of $G_{i,j}$ will yield to another proper-coloring of $G_{i,j}$. Then $G_{i,j}$ can either contain one of v_i and v_j or both of them. If $G_{i,j}$ only contains v_i, then we can swap the colors i and j in that component obtaining a new proper 5-coloring of $G - v$ such that the neighbors of v do not have color i. This means G is 5-colorable, unless for each i and j in $G_{i,j}$ it include both v_i and v_j.

Now we define $P_{i,j}$ to be the path from v_i to v_j in $G_{i,j}$, demonstrated as an example below as $(i, j) = (1, 3)$ in Figure 3.15. Adding v to the path $P_{1,3}$ will make the cycle C, which will separate v_2 from v_4. According to Theorem 19, path $P_{2,4}$ must cross C. Although as G is planar, $P_{2,4}$ and C can cross only at their shared vertices. Path $P_{2,4}$ only include vertices with color 2 and 4, as well path $P_{1,3}$ include vertices of color 1 and 3, which means this two paths do not cross each other, which is a contradiction. We can conclude that it is not possible for $G_{i,j}$ to include both v_i and v_j, so after all we can conclude that the graph G is 5-colorable.

□

Although nowadays a graph being 5-colorable is an important feature which has many applications in different fields, at the time it was questioned by mathematicians if being 5-colorable was enough to expect from a planar graph or not, which has yield to a new

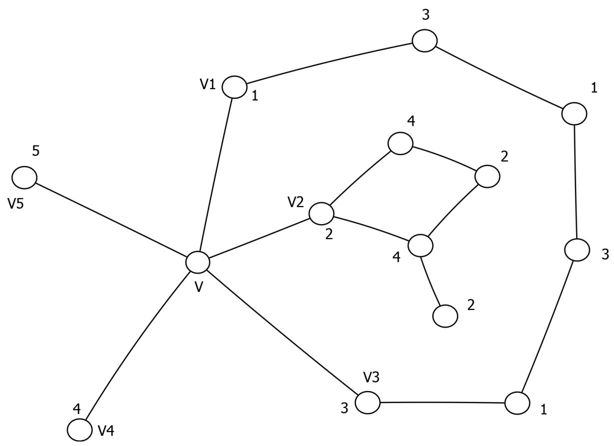

Figure 3.15. *An example for cycle C, which will be completed by $P_{1,3}$*

theorem about coloring of planar graphs.

Theorem 39 (Four Color Theorem)**.** *Every planar graph is 4-colorable [11, p. 260].*

Using same induction method to prove Theorem 39 will cause some difficulties, resulting to a complicated algorithm with an unavoidable set of 633 non reducible configuration. Although the induction method was firstly used to prove Theorem 39 by a computer program, the complexity of the proof resulted the need to develop a new approach to prove it which can be studied in [11].

As it was shown in Example 6, Petersen graph has a K_5 minor, which based on Kuratowski's Theorem 23, it is not a planar graph. But it was shown in Example 35 that it is a 4-colorable. This can serve as an example showing that although all planar graphs are 4-colorable, some non-planar graphs can be 4-colorable as well.

4. CONCLUSION

The aim of this thesis was to give an introduction to planarity of graphs. In particular, the results that can be used to test if a graph can be planar or not were introduced and a planarity testing algorithm was given.

The Kuratowski's theorem states that a given graph is planar if and only if it does not contain K_5 or $K_{3,3}$. The algorithm was developed based on Kuratowski's theorem to determine planarity of a graph and to represent a planarity embedding of a given graph if it has any, which showed that every graph without a K_5 and $K_{3,3}$ is a planar graph. Also if a graph has K_5 or $K_{3,3}$ minor then it is not planar, as they are not planar themselves. Later a simple method was introduced to approve if a given graph is not planar based on Euler formula.

Using Euler's Formula and its result we showed that K_5 or $K_{3,3}$ graphs are not planar. Another result of Euler's Formula was discussed, showing that in a planar graph, number of edges is bounded to the number of vertices, which later was used in the proof of 5-coloring properties of planar graphs.

Then, as graph coloring has a wide range of application as it can classifies vertices of a graph into some subcategory, the coloring properties of planar graphs were defined. It was proofed that planar graphs are 5-colorable. Also, it was mentioned that it is possible to show that palanar graphs have a 4-coloring embedding. So, even though some non-planar graphs are 4-colorable, all planar graphs are 4-colorable.

In this work we studied one algorithm that identified planarity of a given graph. There exist several other planarity testing algorithms besides the one presented here. For example, the algorithm presented in [12, p.135–158] by J. Hopcroft that works in linear time. Although, we have observed that planarity of graphs is a well-studied topic the work on planarity algorithms is still active [13].

REFERENCES

[1] Dong, W. L (p, q)-labeling of planar graphs with small girth. *Discrete Applied Mathematics* 284 (2020), pp. 592–601.

[2] Jepsen, T. S., Jensen, C. S. and Nielsen, T. D. Relational Fusion Networks: Graph Convolutional Networks for Road Networks. *IEEE Transactions on Intelligent Transportation Systems* (2020).

[3] Guinard, S., Landrieu, L., Caraffa, L. and Vallet, B. Piecewise-Planar Approximation of Large 3D Data as Graph-Structured Optimization. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 4 (2019).

[4] Çaylak, O., Lilienfeld, O. A. von and Baumeier, B. Wasserstein metric for improved quantum machine learning with adjacency matrix representations. *Machine Learning: Science and Technology* 1.3 (2020), 03LT01.

[5] Michail, D., Kinable, J., Naveh, B. and Sichi, J. V. JGraphT—A Java Library for Graph Data Structures and Algorithms. *ACM Transactions on Mathematical Software (TOMS)* 46.2 (2020), pp. 1–29.

[6] Patrignani, M. *Planarity Testing and Embedding*. CRC Press, 2004.

[7] Bondy, J. A. *Gratuate texts in Mathematics: Graph Theory*. Springer, 2008. 651 p.

[8] Mann, M. P., Zion, B., Shmulevich, I., Rubinstein, D. and Linker, R. Combinatorial Optimization and Performance Analysis of a Multi-arm Cartesian Robotic Fruit Harvester—Extensions of Graph Coloring. *Journal of Intelligent & Robotic Systems* 82.3-4 (2016), pp. 399–411.

[9] Schlueter-Kuck, K. L. and Dabiri, J. O. Coherent structure colouring: identification of coherent structures from sparse data using graph theory. *Journal of Fluid Mechanics* 811 (2017), pp. 468–486.

[10] Sadati Tileboni, S. A., Jazayeriy, H. and Valinataj, M. Genetic Algorithm with Intelligence Chaotic Algorithm and Heuristic Multi-Point Crossover for Graph Coloring Problem. *Signal and Data Processing* 14.2 (2017), pp. 75–96.

[11] West, D. B. *Introduction to graph theory*. Pearson Modern Classic, 2018. 365 p.

[12] Hopcroft, J. *Dividing a graph into triconnected components*. Comput, 1973.

[13] Didimo, W., Liotta, G. and Patrignani, M. HV-planarity: Algorithms and complexity. *Journal of Computer and System Sciences* 99 (2019), pp. 72–90.