Regular Augmentations of Planar Graphs with Low Degree

Bachelor Thesis of

Huyen Chau Nguyen

At the Department of Informatics
Institute of Theoretical Computer Science

Reviewers: Prof. Dr. Dorothea Wagner
Prof. Dr. Peter Sanders

Advisors: Dipl.-Inform./Dipl.-Math. Tanja Hartmann
Dr. Ignaz Rutter

Time Period: 1st June 2013 – 30th September 2013

Statement of Authorship

I hereby declare that this document has been composed by myself and describes my own work, unless otherwise acknowledged in the text.

Karlsruhe, 27th September 2013

Abstract

In this work we present solutions on how to augment a planar, 2-regular graph such that the resulting graph becomes 3-regular while remaining planar. Furthermore, we describe different augmentations in regard of connectivity, 2-connectivity or 3-connectivity of the resulting graph with either a variable or fixed embedding. We show that the decision problem as well as finding an augmentation can be computed in linear time.

Deutsche Zusammenfassung

In dieser Arbeit präsentieren wir Lösungen, wie man einem 2-regulären, planaren Graph Kanten hinzufügen kann, sodass der Graph planar bleibt und 3-regulär wird. Zusätzlich betrachten wir den Zusammenhang der resultierenden Graphen und stellen verschiedene Lösungen für einfach, zweifach und dreifach zusammenhängende Graphen mit variabler oder fester Einbettung vor. Wir zeigen, dass das Entscheidungsproblem sowie das Finden einer Augmentierung in linearer Zeit lösbar ist.

Contents

1. Introduction

Planarity as well as regularity are strong features of graphs. Problems such as MIXED-MAX-CUT which are NP-hard for general graphs, can be solved in $O(n^{\frac{3}{2}} \log(n))$ for planar graphs [SWK90]. Rosenfeld [Ros64] shows that because of the uniform degree distribution of regular graphs, information such as upper bounds of independent sets are known for regular graphs which are unknown in general graphs.

Therefore, planarity and regularity are features of a graph that are interesting to keep or obtain. In this work we will focus on the problem of augmenting a 2-regular planar graph by adding edges such that it becomes 3-regular while remaining planar. Furthermore, the connectivity of a graph indicates how robust a graph is. Connectivity is therefore another interesting feature. A 3-regular graph can be at most 3-connected as every single vertex can be isolated by removing three edges. We will therefore additionally examine how to augment a graph such that the graph becomes c-connected for $c \in \{0, 1, 2, 3\}$. A c-connected augmentation with $c = 0$ means that the resulting graph does not need to be connected. An *augmentation* of a graph $G = (V, E)$ is a set $W \subseteq E^c$ of edges of the complement graph. The *augmented graph* $G' = (V, E \cup W)$ is denoted by $G + W$ and is simply the graph G with additional edges that have not existed before.

Nagamochi and Ibaraki [NI02] examine how to augment a graph such that it becomes 2-connected. Taking planarity into account, Kant and Bodlaender [BKK+91] show that the problem how to find an augmentation W of a planar graph $G = (V, E)$ such that $G+W$ becomes 2-connected and remains planar is NP-hard. They present two approximation algorithms for this problem. Hartmann, Rollin and Rutter [HRR12] show that finding such an augmentation and additionally demanding 3-regularity is NP-hard for graphs with a variable embedding. If the embedding is fixed, the problem is solvable in $O(n^{2.5})$. In the current work of Hartmann, Rollin and Rutter [HRR13] an algorithm is presented that solves the problem in $O(n^{1.5})$. However, finding a planar, 3-regular and 3-connected augmentation for graphs with a fixed embedding (this problem is denoted by 3-FEPRA) is again NP-hard. Also, finding planar, 3-regular and c-connected augmentations for a graph with a variable embedding is NP-hard for $c \in \{0, 1, 2, 3\}$. This problem is denoted by c-connected 3-PRA.

The problems 3-PRA and 3-FEPRA can be solved in linear time when constraining the given instances. In this work we can show that we can find a planar, 3-regular and c-connected augmentation for 2-regular graphs in linear time for $c \in \{0, 1, 2, 3\}$ regardless whether a graph has a variable or a fixed embedding. We will solve the following problems in this work.

Problem: Planar 3-Regular Augmentation of 2-Regular Graphs (PRA-2G)
Instance: Planar, 2-regular graph $G = (V, E)$ with variable embedding
Task: Find an augmentation W such that $G + W$ is planar and 3-regular.

Problem: Fixed-Embedding Planar 3-Regular Augmentation of 2-Regular Graphs (FEPRA-2G)
Instance: Planar, 2-regular graph $G = (V, E)$ with a fixed embedding
Task: Find an augmentation W such that $G + W$ is planar and 3-regular while preserving the given embedding.

This work shows that by only dealing with 2-regular graphs, c-connected PRA-2G is solvable for all graphs with an even number of vertices for $c \in \{0, 1, 2, 3\}$. Whether a graph has an even number of vertices can be checked in linear time (and thus deciding whether a graph admits a planar, 3-regular and c-connected augmentation) in contrast to 3-PRA which is NP-hard. The problem FEPRA-2G is also solvable in linear time in contrast to c-connected 3-FEPRA which is NP-hard for $c = 3$ or solvable in $O(n^{1.5})$ for $c \in \{0, 1, 2\}$.

In order to solve FEPRA-2G, we first examined 3-regular augmentations for tunnels which are 2-regular graphs with a fixed embedding where each face is incident to at most two connected components. We summarize our results for c-connected FEPRA-2G in Table 1.1 and show which tunnels and which graphs admit an augmentation for c-connected FEPRA-2G. As any 3-regular graph must have an even number of vertices (see Lemma 2.1) this is also a necessary condition for the input graphs. Thus, we only consider graphs with an even number of vertices as input graphs in Table 1.1. In most cases a specific characterisation is given. We proved that those characterisations can be checked in linear time. In the case of connected FEPRA-2G we could not find a characterisation but provide a linear algorithm instead, which computes an augmentation in linear time.

Table 1.1.: Overview of planar 2-regular graphs with a fixed embedding that admit a planar, 3-regular and c-connected augmentation

	Tunnel	**FEPRA-2G**
0-connected	The tunnel must not be a TST-containing tunnel. See Theorem 4.3.	The graph must not be a TST-containing graph. See Theorem 5.8.
1-connected	The tunnel 1. must be either a 3-3 tunnel or 2. must not start or end with two triangles or 3. must not be a TST-containing graph. See Theorem 4.2.	We provide no characterisation for graphs that admit a connected augmentation. Instead, we provide an algorithm which checks whether the graph admits such an augmentation in linear time. See Algorithm 5.1.
2-connected	The tunnel 1. must not have a triangle as a middle cycle and 2. all of its cycles that have an odd number of vertices inside must consist of at least five vertices. See Theorem 4.5.	The graph 1. must not have a triangle as a middle cycle and 2. all of its cycles that have an odd number of vertices inside must consist of at least five vertices. See Theorem 5.2.

<table>
<tr><td>3-connected</td><td>The tunnel
1. must have middle cycles consisting of at least six vertices and
2. all of its middle cycles that have an even number of vertices inside must consist of at least seven vertices.
See Theorem 3.11.</td><td>The graph
1. must have middle cycles consisting of at least six vertices and
2. all of its middle cycles that have an even number of vertices inside must consist of at least seven vertices.
See Theorem 5.4.</td></tr>
</table>

Outline. The preliminary in Section 2 provides all basic concepts and notations that are needed for this work. In Section 3, we examine PRA-2G for all possible connectivity constraints and the time complexity of these problems. In Section 4, we introduce a representation for 2-regular graphs with a fixed embedding and we examine how to augment tunnels such that they become planar and 3-regular in all aspects of connectivity. Afterwards, we provide solutions for c-connected FEPRA-2G in Section 5 for $c \in \{0, 1, 2, 3\}$ and we analyse their time complexity.

2. Preliminaries

A graph $G = (V, E)$ is called *k-regular* if all vertices $v \in V$ have degree k. A graph G is planar if it can be embedded in a plane in such a way that no edges are crossing each other. The embedding of a graph divides the plane into different faces. There is always an outer face which is the face that encloses the embedded graph. A planar graph can always be embedded in such a way that a fixed face f can become the outer face. This is obvious when imagining the embedding drawn onto a sphere and rolled out on a plane such that f is the outer face.

An *augmentation* of a graph $G = (V, E)$ is a set $W \subseteq E^c$ of edges of the complement graph. The *augmented graph* $G' = (V, E \cup W)$ is denoted by $G + W$ and is simply the graph G with additional edges that have not existed before. Every vertex of a 2-regular graph G has degree 2. To become 3-regular, each vertex requires one additional incident edge in the augmentation W to achieve degree 3. We therefore call such a vertex a *valency*. An edge connects two vertices and satisfies the demand of two vertices. Hence, the following Lemma is true.

Lemma 2.1. *A graph that admits a regular augmentation must have an even number of vertices.*

A graph $G = (V, E)$ is *k-edge-connected* if it remains connected after removing less than k edges. This means that in order to separate two vertices u and v, at least k edges have to be removed. An edge-cut of two vertices u and v is a set of edges that separates the two vertices if the edge-cut is removed. Thus, the minimum size of an edge-cut of two vertices u and v in a k-connected graph is at least k. *Menger's Theorem* says that the size of a minimum edge-cut of two vertices u and v is equal to the maximum number of edge-disjoint paths between u and v. The size of a minimum edge-cut between u and v is denoted by $\lambda(u, v)$. Similarly, a connected graph $G = (V, E)$ is *k-vertex-connected* if $k < |V|$ and if G remains connected after deleting less than k vertices. In order to separate two nonadjacent vertices u and v of a k-vertex-connected graph, k vertices have to be removed and Menger's Theorem says that the size of the minimum vertex-cut of two non-adjacent vertices u and v is equal to the maximum number of vertex-disjoint paths between u and v. The size of the minimum vertex-cut between u and v is denoted by $\kappa(u, v)$.

Though edge-connectivity and vertex-connectivity are different, they are equal for graphs with maximum degree 3.

Lemma 2.2. *Let graph $G = (V, E)$ be a max 3-degree graph. The edge-connectivity of G and the vertex-connectivity of G are equal.*

Proof. Let u and v be two vertices of graph G and let G be k-edge-connected. As G is k-edge-connected, there are k edge-disjoint paths $p_1, p_2, \ldots, p_k$ between u and v. Since G is a graph with maximum degree 3, there is no vertex $w \in V \backslash \{u, v\}$ that is incident to edges of two distinct paths p_i and p_j with $i \neq j$. Because if there was a vertex w that is part of p_1 ad p_j, w must be incident to the edges $s_1 \in p_i$ and $s_2 \in p_i$ and incident to the edges $t_1 \in p_j$ and $t_2 \in p_j$ which are all distinct. This is not possible because the maximum number of incident edges of w is three.

This means that in order to find a minimum vertex-cut between u and v the k paths $p_1, p_2, \ldots p_k$ have to be disconnected by deleting vertices of these paths. There is no vertex that can disconnect multiple paths at once and we have to delete k vertices to disconnect the k paths. Thus, the vertex connectivity of G is also k.

This shows that the vertex-connectivity is k if the edge-connectivity for a graph with maximum degree 3 is k. As the edge-connectivity of a graph is always bigger or equal the vertex-connectivity, the vertex-connectivity and the edge-connecticity are always equal for graphs with maximum degree 3. □

Furthermore, we will now show that edge connectivity is transitive.

Lemma 2.3. *Let $G = (V, E)$ be a graph and a, b, $c \in V$ vertices of G. For the vertices a, b and c, it is $\lambda(a, b) = 3$ and $\lambda(b, c) = 3$. Then $\lambda(a, c) = 3$.*

Proof. Let edge-cut C be the smallest cut separating vertex a from c. Now there are two possibilites. Either vertex b is still in the same component as a. That would mean, that C separates b and c. As it is $\lambda(b, c) = 3$, $|C|$ must be at least three. Else if b is not in the same component as a, cut C is a cut that separates the vertices a and b which means that $|C|$ must also be at least three. That means $\lambda(a, c) = 3$. □

The distance of two vertices u and v in a connected graph is the number of edges in a shortest path connecting u and v. In a connected graph that contains two distinct paths P and L, the *distance of P and L* is $k = \min\{\text{distance}(u, v) \mid u \in P \wedge v \in L\}$.

A 2-regular connected component that is a cycle consisting of x vertices, is denoted by a *x-cycle.*

We denote a graph with a fixed embedding by a *tunnel* if each face of the graph is incident to at most two connected components. That means that we have cycles that are nested in each other. We call a tunnel a *x-y-z* tunnel if the graph consist of three cycles consisting of x, y or z vertices. The outermost cycle consists of z vertices, enclosing another cycle consisting of y vertices which is enclosing the third cycle consisting of x vertices.

3. Planar 3-Regular Augmentation of 2-Regular Graphs

In this section we show how to find a cubic augmentation of a 2-regular (and planar) graph, such that it remains planar. We will study the following two problems.

Problem: Planar 3-Regular Augmentation of 2-Regular Graphs (PRA-2G)
Instance: Planar, 2-regular graph $G = (V, E)$ with variable embedding
Task: Find an augmentation W such that $G + W$ is planar and 3-regular.

Problem: Fixed-Embedding Planar 3-Regular Augmentation of 2-Regular Graphs (FEPRA-2G)
Instance: Planar, 2-regular graph $G = (V, E)$ with a fixed embedding
Task: Find an augmentation W such that $G + W$ is planar and 3-regular while preserving the given embedding.

The connectivity of a graph indicates how robust a graph or network is. On that account we will also study the problem of *c-connected* PRA-2G and c-connected FEPRA-2G for $c = 1, 2, 3$, where an augmentation is sought, such that the resulting graph is additionally c-connected.

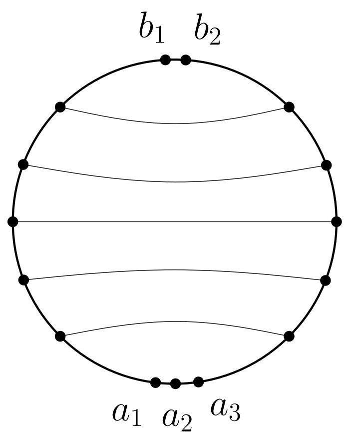


Figure 3.1.: An example of a cycle processed with Rule 3.1 except five vertices.

As the given instance is a 2-regular graph, the graph can only consist of the following three different types of components:

1. **Even cycles:** A component that is a simple cycle consisting of n vertices, where n is even and $n \geq 4$.
2. **Odd cycles:** A component that is a simple cycle consisting of n vertices, where n is odd and $n \geq 3$.
3. **Triangle:** A triangle is a special odd cycle that consists of three vertices.

Before examining the problems we show that it is possible to process a cycle $C = (V, E)$ in such a way that the resulting graph is planar and all vertices have degree 3 except for k vertices, where $2 \leq k \leq |V|$ and $|V| - k$ is even. We first draw the cycle in a planar way such that it now has an inner face and an outer face.

Rule 3.1 (Processing a cycle except k vertices)**.**

1. *Choose* $\lceil k/2 \rceil$ *arbitrary vertices* $\{a_1, a_2, \dots a_{\lceil k/2 \rceil}\}$ *that form a path.*
2. *Choose* $\lfloor k/2 \rfloor$ *vertices* $\{b_1, b_2, \dots b_{\lfloor k/2 \rfloor}\}$ *that form a path and have the largest distance to the path* $\{a_1, a_2, \dots a_{\lceil k/2 \rceil}\}$*.*
3. *Process all remaining vertices by connecting the pairs of vertices that have the same distance to the compound vertex* $\{a_1, a_2, \dots a_{\lceil k/2 \rceil}\}$ *.*

An example is shown in Fig. 3.1. This augmentation is planar and all vertices have degree 3 except five vertices. With this rule we will show in the following that all even cycles can be fully processed as well as odd cycles except for one vertex.

We can apply Rule 3.1 except for two vertices a_1 and b_1 on an even cycle. The two vertices that still have two valencies can now be connected by an edge in the outer face as in Fig. 3.2. The resulting graph is planar, 3-regular and even 3-connected. Thus, we can formulate the following Lemma.

Lemma 3.2. *Any even cycle can be augmented in such a way that it becomes planar, 3-regular and 3-connected.*

Processing an odd cycle can be done in a similar way. A cycle with an odd number of vertices cannot become 3-regular, because it has an even number of vertices. But we can show that we can process an odd cycle such that all vertices have degree 3 except for one degree-2 vertex.

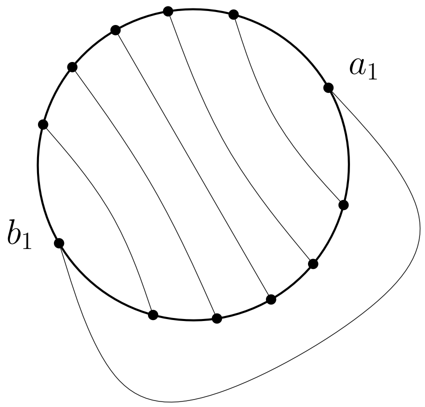


Figure 3.2.: An example of a fully processed even cycle.

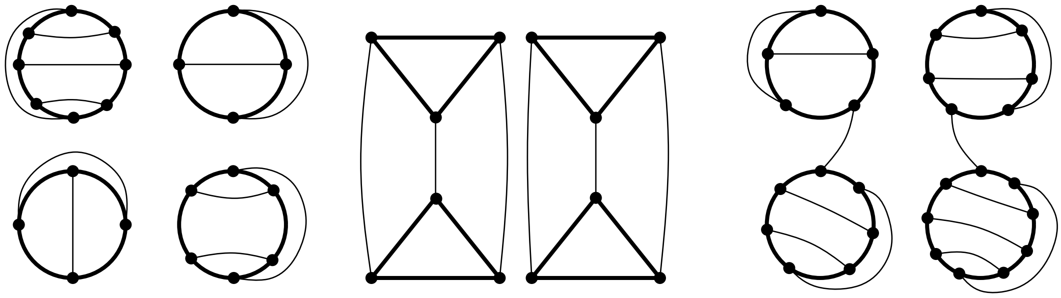


Figure 3.3.: An example of a processed graph using Theorem 3.4.

Lemma 3.3. *Any odd cycle that is no triangle can be augmented in such a way that it becomes planar and 3-regular except for one vertex that is incident to the outer face.*

Proof. An odd cycle can be augmented with Rule 3.1 except for three vertices. The cycle then has two adjacent vertices a_1, a_2 and another vertex b_1 with degree 2. We connect the vertices a_1 and b_1 as in Lemma 3.2. The edge connecting a_1 and b_1 can be embedded in two ways, starting from vertex a_1 and surrounding the cycle either clockwise or anticlockwise. We embed the edge in such a way that the vertex a_2 remains incident to the outer face. The resulting graph is planar and 3-regular except for the one vertex a_2. □

3.1. PRA-2G Without Regarding Connectivity

With these observations we show in the following that any 2-regular graph consisting of an even number of vertices can be augmented in such a way that it becomes planar and 3-regular if the embedding is variable.

Theorem 3.4. *PRA-2G is solvable if the number of vertices in the graph is even.*

Proof. To show that PRA-2G is solvable for a graph G if the number of vertices is even, we examine even cycles, triangles and odd cycles that are no triangle separately. As the embedding is variable, all cycles can be drawn next to each other in a planar way such that no cycles are nested.

We first augment all even cycles using Lemma 3.2. All even cycles are now planar and 3-regular and need not be regarded any further.

Now we aim at processing all triangles. Two triangles abc and $a'b'c'$ can be augmented in a planar way by adding the edges $\{a, a'\}$, $\{b, b'\}$ and $\{c, c'\}$. This way we can augment all triangles if there is an even number of triangles. If there is an odd number of triangles there will be one triangle $x_1x_2y_1$ left that can not be processed yet. But then, the number of the remaining odd cycles must be odd (as the sum of all vertices is even and thus the number of odd cycles (including triangles) must be even). We choose an arbitrary odd cycle and apply Rule 3.1 except for three vertices a_1, a_2 and b_1. We connect the triangle that has no partner with this odd cycle by adding the edges $\{x_1, a_1\}$, $\{x_2, a_2\}$ and $\{y_1, b_1\}$. Now all triangles (and possibly one odd cycle) are planar and 3-regular and do not have to be regarded further.

Now there is only an even number of odd cycles left and can be processed as follows. We augment all odd cycles as described in Lemma 3.3. Now all vertices have degree 3 except one vertex of each odd cycle. A pair of odd cycles can then be processed by connecting the two vertices that still have one valency left.

After processing all even cycles, triangles and odd cycles, the whole graph is still planar and 3-regular. □

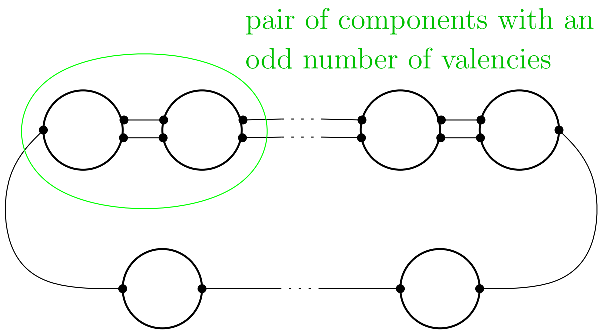


Figure 3.4.: A schematization of a 2-connected augmentation. Cycles with three dots indicate components with three valencies. Cycles with two dots indicate components with two valencies.

Figure 3.3 shows an augmentation of a graph using Theorem 3.4.

3.2. 2-Connected PRA-2G

As we complete each component as separately as possible in the previous section, the resulting graph is most likely not connected. In this section we show that all graphs with an even number of vertices can be augmented in such a way that the resulting graph is 2-connected. Later, we will see that we can even achieve 3-connectivity.

In a first step we assume a graph that consists of several connected components that are already 2-connected but still provide some valencies. Later, we will use this lemma to prove 2-connected PRA-2G and 2-connected FEPRA-2G.

Lemma 3.5. *Let $G = (V, E)$ be a planar graph that consists of multiple connected components that are 2-connected. Each of these components consist of vertices that have degree 3 except for either two or three vertices with degree 2. Furthermore, all vertices with degree 2 are incident to a common face. The graph G can be augmented such that it becomes planar, 3-regular and 2-connected if the number of vertices is even.*

Proof. All components that have three valencies can be connected in pairs by two edges resulting in a new 2-connected component with two valencies. This is possible because there must be an even number of components with three valencies as a graph has always an even sum of vertex degrees because an edge always increases the vertex degrees of a graph by 2. The graph G has an even number of vertices which have either degree 2 or degree 3. Thus, there must be an even number of vertices with degree 3 and also an even number of vertices with degree 2. Hence, there must be an even number of components with three valencies.

Now, all connected components have two valencies that are incident to the same face. All these components can now be connected in a ring as in Fig. 3.4.

In order to separate two vertices that are part of the same component C, at least two edges need to be deleted as the component C is 2-connected. In order to separate two vertices that are part of two different components, at least two edges need to be deleted, because the components are connected in a ring. Hence, the resulting graph is 2-connected. □

Theorem 3.6. *All 2-regular graphs that consist of an even number of vertices can be augmented such that the resulting graph is planar, 3-regular and 2-connected.*

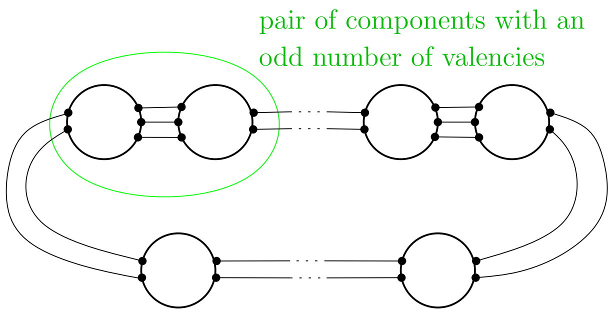


Figure 3.5.: A schematization of a 3-connected augmentation

Proof. If the graph is one even cycle, we find a planar, 3-regular and 2-connected augmentation by using Lemma 3.2.

Else, all even cycles can be augmented with Rule 3.1 except for two vertices. All odd cycles can be augmented with the same rule except for three vertices (triangles do not need to be further augmented). This results in cycles that are almost 3-regular aside from two or three vertices. We now draw all cycles in a planar way such that the remaining valencies are incident to the same face. As cycles are 2-connected, we now have fulfilled all conditions for Lemma 3.5. The graph can therefore be augmented such that it becomes planar, 3-regular and 2-connected. □

3.3. 3-Connected PRA-2G

In this section, we show that it is not only possible to find an augmentation that makes the resulting graph 2-connected, but even 3-connected. In order to show that G can be augmented such that it becomes planar, 3-regular and 3-connected, we first find a valid augmentation for graphs that only consist of even and odd cycles that are no triangles. Then we show that we find an augmentation for graphs that only consist of triangles. Afterwards, we show that we can find a combined augmentation out of these two cases to find a valid augmentation for any 2-regular graph such that the resulting graph is planar, 3-regular and 3-connected.

Lemma 3.7. *Let $G = (V, E)$ be a planar graph that consists of multiple connected components C_1, C_2, ..., C_n that are 2-connected. Each of these components consist of vertices that have degree 3 except for either four or five vertices with degree 2. Furthermore, all vertices with degree 2 are incident to a common face. Graph G can then be augmented such that it becomes planar and 3-regular and 2-connected if the number of vertices is even.*

Proof. This proof is similar to the proof of Lemma 3.5. If component C_i has five valencies, they are clockwise named s_i, t_i, u_i, v_i and w_i. A pair of components with five valencies C_i and C_k can now be fused by adding the edges (u_i, u_k), (v_i, v_k) and (w_i, w_k). This new component now has four valencies left, provided by the vertices s_i, s_k, t_i and t_k.

Now, each connected component has four valencies. They can be connected in a ring by always connecting two components by two edges as shown in Fig. 3.5.

Let A and B be two neighbouring components in the ring. As every component has at least two connections to each of its neighbours, A has two edges that connect A with B. These both edges are two edge-disjoint paths. A third path connecting A and B can be found by tracing the component ring counterclockwise until cycle B is reached. Thus $\lambda(A, B) = 3$. As 3-connectivity is transitive for 3-regular graphs, all components are 3-connected among each other.

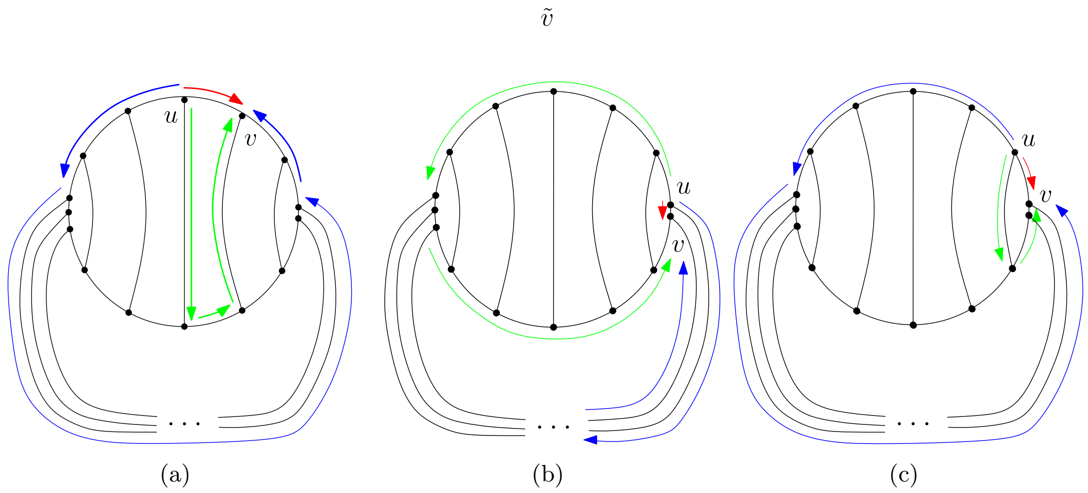


Figure 3.6.: Figure (a) illustrates three edge-disjoint path between two vertices, if both of them are not connected to another component, Figure (b) illustrates paths, if both vertices are connected to another component, Figure (c) illustraties paths, if only one vertex is connected to another component.

As the connected components were 2-connected before, the whole graph G is now planar, 3-regular and 3-connected. □

Lemma 3.8. *A 2-regular graph that does not contain triangles can be augmented such that the resulting graph is planar, 3-regular and 3-connected if the number of vertices is even.*

Proof. If the whole graph consists of one even cycle, we find a planar, 3-regular and 3-connected augmentation by using Lemma 3.2.

Else, we augment all even and odd cycles with Rule 3.1 except for four or five vertices. We now draw all cycles in a planar way such that all remaining valencies are incident to the same face. Now, the conditions for Lemma 3.7 are fufilled and there exists an augmentation such that the graph becomes planar, 3-regular and 2-connected.

In fact, the resulting graph is even 3-connected which we will show in the following. As shown in Lemma 3.7, for two components A and B it is $\lambda(A, B) = 3$. What is left to show is, that for two adjacent vertices u and v of one former cycle C', it is $\lambda(u, v) = 3$. We show this by finding three edge-disjoint paths between u and v. Without loss of generality we assume that v is the next vertex that is found from u when following the cycle in clockwise direction.

If both u and v are not directly connected to another component, the three paths are shown in Fig. 3.6 (a) and they can be found as follows. The first path is $\{u, v\}$. The second path is from u following the cycle anticlockwise until the ring is reached. From the ring we reach the cycle from the other side and follow the cycle anticlockwise to reach v. The third path is starting at u following the incident edge, that was augmented with Rule 3.1. From there, we reach the other side of the cycle which we follow anticlockwise to the next vertex. This vertex is adjacent to v and we can therefore reach v.

If both u and v are directly connected to another component, the three paths are shown in Fig. 3.6 (b). The edge $\{u, v\}$ is the first path. The second path starts at u following the cycle anticlockwise until v is reached. The third path starts at u following the edge that connects to another component C''. From C'' there must be a vertex that is connected to v and v is reachable. □

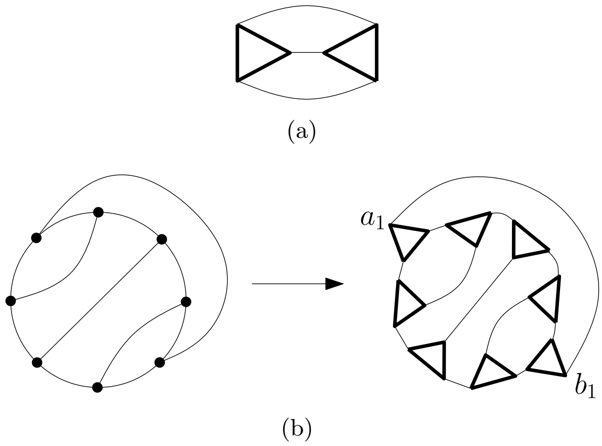


Figure 3.7.: 3-connected augmentations of 2 and 8 triangles.

If only one of both vertices is connected to another component, we consider this vertex to be vertex v and the one vertex that is not connected to another component is u. Figure 3.6 (c) shows the three paths. The first path is the edge $\{u, v\}$. The second path starts at u following the cycle anticlockwise until the ring of component can be reached. From there we can reach the component, that is connected to v and reach v. The third path starts at u and follows the edge that was augmented with Rule 3.1. From there we follow the cycle anticlockwise and reach v. Thus, the augmentation is planar, 3-regular and 3-connected.

Lemma 3.9. *A 2-regular graph G consisting of an even number of vertices, whose connected components are triangles, can be augmented such that the resulting graph is planar, 3-regular and 3-connected.*

Proof. Let n be the number of triangles that are in graph G. Then n must be an even number, because the number of vertices is even. If $n > 2$ we construct a graph G' that is a cycle consisting of n vertices augmented by using Lemma 3.2. The resulting graph is planar, 3-regular and 3-connected.

Now we replace each vertex s in C with a triangle abc. The incident edges to s, called $\{s, u\}$, $\{s, v\}$ and $\{s, w\}$, are removed and instead we add $\{a, u\}$, $\{b, v\}$ and $\{c, w\}$. The resulting graph equals an augmented G that is planar, 3-regular and 3-connected, as we connect every vertex of a triangle with another vertex.

As a triangle provides three valencies from three different vertices, we can even process a graph consisting of two triangles abc and $a'b'c'$ by adding the edges $\{a, a'\}$, $\{b, b'\}$ and $\{c, c'\}$. Thus, we find a planar, 3-regular and 3-connected augmentation for all graphs consisting of an even number of triangles. Fig. 3.7 (a) shows a 3-connected augmentation for two triangles and (b) a 3-connected augmentation for eight triangles. □

As it is possible to find a valid augmentation for graphs without triangles and a valid augmentation for graphs consisting of triangles, we now need an augmentation for a graph consisting of all kinds of cycles. In order to do this we first show how a vertex can be added to a 3-connected graph such that the graph still remains 3-connected.

We consider a graph $G = (V, E)$ that is 3-connected. Let $\{s, t\}$, $\{u, v\} \in E$ be two disjoint edges that are incident to the same face f. Then we can add a vertex w in face f, remove the edges $\{s, t\}$ and $\{u, v\}$ and instead add the edges $\{s, w\}$, $\{t, w\}$, $\{u, w\}$ and $\{v, w\}$.

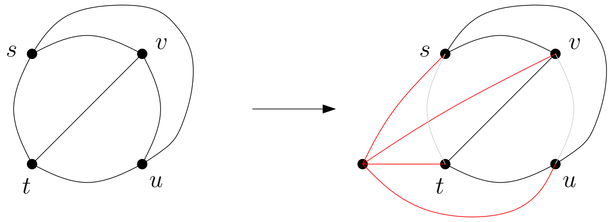


Figure 3.8.: This figure shows how to add a vertex to the K_4-graph by splitting two edges, as the K_4 does not have four vertices that are incident to a common face.

Lemma 3.10. *$G + w$ is still 3-edge connected.*

Proof. As G was 3-connected, there were three edge-disjoint paths from any vertex to any other. As the construction splits edges but does not reduce the number of paths between vertices, there are still three edge-disjoint paths from each vertex in G to every other vertex in G. As 3-edge-connectivity is transitive by Lemma 2.3, what is left to show is that $\lambda(s, w) = 3$ in $G + w$.

Let C be a minimal cut that separates w and s. If C also separates w from t, u or v, then it is $|C| \geq 4$, because it separates w from its four neighbours. If C does not separate w from one of the vertices, without loss of generality we say that C does not separate w and t, then C separates w and s. This would mean that C separates s and t and $|C| \geq 3$ because s and t were 3-edge-connected. As $|C| \geq 3$, it is $\lambda(s, w) = 3$. □

Theorem 3.11. *A graph $G = (V, E)$ consisting of an even number of vertices can be augmented such that the resulting graph is planar, 3-regular and 3-connected.*

Proof. Lemma 3.8 and Lemma 3.9 show that graphs that only consist of triangles and graphs that contain no triangles can be processed such that the resulting graph is planar, 3-regular and 3-connected. We now describe an augmentation for a graph that consists of even and odd cycles (including triangles). If G consists of an even number of triangles, we process all triangles as described in Lemma 3.9.

Else if the number of triangles is odd, process additionally one odd cycle with Rule 3.1 except three vertices. Now this cycle can be treated just like a triangle and the triangles plus this cycle can be processed according to Lemma 3.9. All remaining cycles are processed according to Lemma 3.8. Now, the graph consists of a ring of cycles R and a structure of triangles Δ. Both connected components are 3-connected.

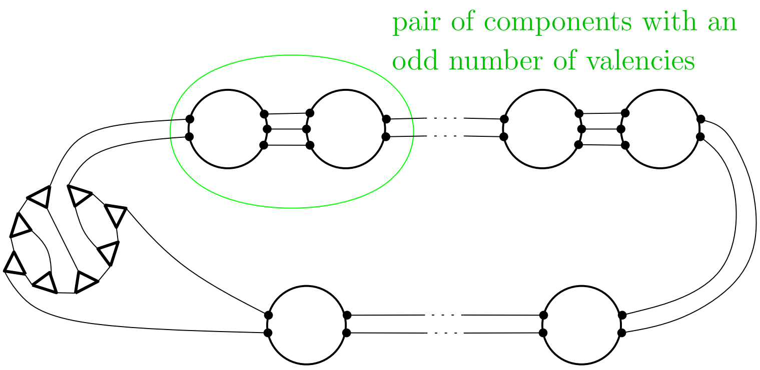


Figure 3.9.: A 3-connected, planar and 3-regular augmentation for a graph with triangles, even and odd cycles.

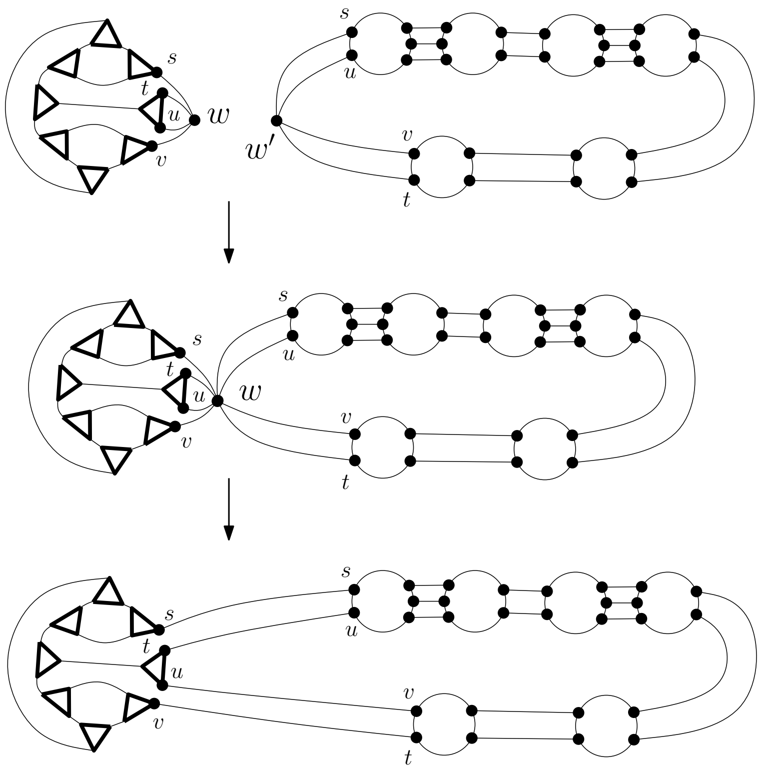


Figure 3.10.: This Figure shows two components, one triangle component and one component without triangles with the added vertices w and w'. Then the vertices are fused to vertex w which is later on removed.

If R is an augmented cycle consisting of four vertices, we can add an additional vertex as shown in Figure 3.8. Otherwise, we add one vertex w to the component R according to Lemma 3.10. Then, we add a vertex w' to the component Δ according to Lemma 3.10. As the components are planar, they can be embedded in such a way that the vertices w and w' lie in the outer face. Both vertices w and w' are therefore incident to a common face. We now identify the vertices w and w' to a vertex w by "melting" both vertices to a vertex w and the two components are then connected. As both components are 3-connected, for all vertices u of Δ it is $\lambda(u, w) = 3$ and for all vertices v of R it is $\lambda(v, w) = 3$ which makes the whole graph 3-edge-connected. The vertices that are adjacent to w and are in the component R, are named s, t, u and v. The vertices that are adjacent to w and are in Δ are named s', t', u' and v'. We now remove vertex w and instead add the edges $\{s, s'\}$, $\{t, t'\}$, $\{u, u'\}$ and $\{v, v'\}$. We now have to prove that the graph is still 3-connected. Fig. 3.10 shows an example.

Let A be a minimal cut that separates vertex s and s'. If C separates s from t', u' and v', $|A|$ must be at least 3, because there are four edges connecting the component R and Δ. Else if C does not separates s from one of these vertices, e.g. with t', A separates s' and t' and $|A|$ must be at least 3, because $\lambda(s', t') = 3$.

The resulting graph is therefore planar, 3-regular and 3-connected. Figure 3.9 shows an example of a connected graph □

3.4. Time Complexity of PRA-2G

Regardless whether considering connected, 2-connected or 3-connected PRA-2G, it only depends on the number of vertices whether the problem is solvable or not. Thus, the decision problem (whether such an augmentation exists), is solvable in linear time.

If we seek the actual augmentation, we first have to find all cycles and the number of vertices it consists of. This can be done by using depth-first-search on the graph. Depth-first-search on a graph $G = (V, E)$ has a time complexity of $O(|V| + |E|)$. As $|E| \in O(|V|)$ for planar graphs, finding all connected components can be computed in linear time.

For the following three theorems, we consider a graph $G = (V, E)$ with k vertices that are part of an even cycle, l vertices that are part of a triangle and m vertices that are part of an odd cycle. The sum of all vertices is even and it is $|V| := n = l + k + m$. We will show that PRA-2G as well as 2-connected and 3-connected PRA-2G is solvable in linear time.

Theorem 3.12. *PRA-2G without regarding connectivity can be solved in $O(n)$ time.*

Proof. When solving PRA-2G, we first process all even cycles, later we process all triangles and odd cycles. When processing an even cycle with a vertices, we use Lemma 3.2 which chooses two vertices u and v that lie opposite to each other in the cycle and connect them by an edge in the outer face and all other vertices can be connected by edges inside the cycle. Augmenting such a cycle needs $O(a)$ time. Thus, processing all even cycles needs $O(k)$ time.

Two triangles abc and $a'b'c'$ can be augmented by adding the edges $\{a, a'\}$, $\{b, b'\}$ and $\{c, c'\}$ and can therefore be done in $O(1)$. If there is an odd number of triangles, one triangle has to be connected to an odd cycle which can also be done in $O(1)$. There are $\frac{l}{3}$ triangles, thus processing all triangles can be computed in $O(l)$.

When augmenting two odd cycles, we use Lemma 3.3 which works similar to Lemma 3.2. All odd cycles are augmented such that every vertex has degree 3 except for one vertex that has degree 2. Two odd cycles with one valency can then be processed by connecting them with an edge. Thus, processing two odd cycles with c vertices needs $O(c)$ time and processing all odd cycles need $O(m)$ time. Thus, solving PRA-2G for graph G, can be done in $O(k + l + m) = O(n)$. □

Theorem 3.13. *The problem 2-connected PRA-2G can be solved in $O(n)$ time.*

Proof. To solve 2-connected PRA-2G, we first augment all even cycles with Rule 3.1 except for two vertices. A cycle with a vertices is augmented by choosing two vertices u and v that lie opposite to each other in the cycle. a pair of two vertices with the same distance to vertex u can be connected by an edge inside the cycle. This way, all vertices have degree 3 except u and v. Augmenting the cycle needs $O(a)$ time. All odd cycles are connected in a ring by connecting all even cycles with an edge. Thus, even cycles can be augmented in $O(k)$.

Odd cycles are augmented by using Rule 3.1 except three vertices. A pair of odd cycles can be connected by two edges and can be appended to the ring. Processing two odd cycles with b vertices can be done in $O(b)$. Thus augmenting all odd cycles can be done in $O(l + m)$ time. 2-connected PRA-2G can therefore be computed in $O(k + l + m) = O(n)$ time. □

Theorem 3.14. *The problem 3-connected PRA-2G can be solved in $O(n)$ time.*

Proof. To solve 3-connected PRA-2G for graph G, we process all triangles by using an adapted Lemma 3.2 which can be done in $O(l)$. All other cycles are processed by using Rule 3.1 which can be computed in $O(k + m)$. To join the component of triangles and the component of other cycles, $O(1)$ time is needed. Thus, 3-connected PRA-2G can be solved in $O(k + l + m) = O(n)$ time. □

4. Planar 3-Regular Augmentation of Tunnels

In the previous chapter we showed how to augment a graph that is 2-regular such that it becomes 3-regular and planar if the embedding is variable. In this section, we discuss such augmentations of 2-regular graphs with a fixed planar embedding. In order to do this, we will first define how to describe a 2-regular graph with fixed embedding.

4.1. Tree Representation of a 2-Regular Graph with Fixed Embedding

As 2-regular planar graphs always consist of cycles, each component is incident to two faces, an inner and an outer face. We can then describe a 2-regular graph by a tree. We add a node for every face and give the node a value. The value of each node indicates the number of vertices of the cycle that is enclosing the face. We identify a node in the tree with the cycle that is enclosing the face for which the node was added. We can therefore for example say that a node is planar or 3-regular. A node A is a child node of another node B if node A lies inside of node B and node A and B are incident to a common face. Thus, a tree representation always has a root with value 0, as there is always an outer

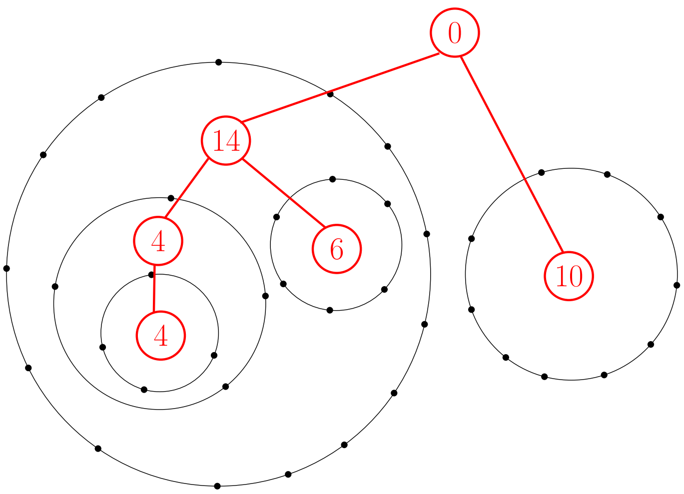


Figure 4.1.: An example of a tree representation of a graph.

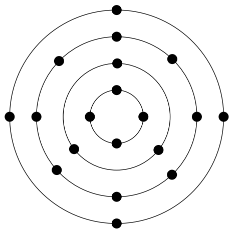


Figure 4.2.: An example of a tunnel

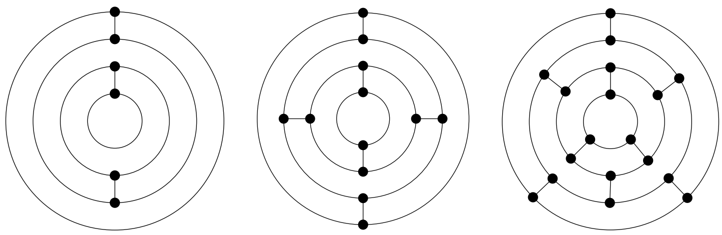


Figure 4.3.: The lower bound of in- and out-edges to make a graph connected (a), 2-connected (b) or 3-connected (c).

face that is not enclosed by a cycle. A subtree of a node K is the subgraph of cycle K and all the cycles that lie inside of K. We can for example say that a subtree is planar or 3-regular. Figure 4.1 shows an example. If the depth of a tree representation is h, the depth of the root is $h-1$ and the depth of the leaf with the lowest depth is zero.

4.2. Planar 3-Regular Connected Augmentation of Tunnels

In order to examine augmentations for fixed graphs, we start by first considering graphs that are tunnels. A *tunnel* is a 2-regular graph whose tree representation is a path. That means that every face is incident to at most two cycles. Fig. 4.2 shows an example of such a tunnel. A tunnel has an *innermost cycle* which is a cycle that is represented by a leaf in the tree representation, meaning that it has no other cycle nested in it. The *outermost cycle* is a cycle that is not nested in another cycle (in the tree representation they are the children of the root). All other cycles are called *middle cycles*. If a cycle A is the child node of cycle B, B is called the *outer neighbour* of cycle A. Vice versa, cycle A is the *inner neighbour* of cycle B. Augmentation edges that connect cycle A with its outer cycle B are *out-edges* of cycle A. Edges that connect cycle B with its inner cycle A are *in-edges* of cycle B. We describe tunnels with the number of vertices of the cycles from the innermost to the outermost cycle. For example, a *x-y-z* tunnel is a tunnel where the innermost cycle has x vertices, its outer neighbour has y vertices and the outermost cycle has z vertices.

When considering a planar, 3-regular and connected augmentation, the following observations are made.

Lower Bound of In- and Out-Edges: If an augmentation for a tunnel is sought that is connected, every middle cycle must have at least one in-edge and one out-edge. The outermost cycle needs to have one in-edge and the innermost cycle needs to have one out-edge. Similarly, when seeking a 2-connected augmentation every cycle needs at least two in- and out-edges or three in- and out-edges if a 3-connected augmentation is sought as seen in Fig. 4.3.

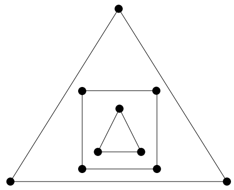


Figure 4.4.: A Triangle-Square-Triangle (TST) Graph

Handing Over Unevenness: A cycle that has as many in- and out-edges as the lower bound for connectivity demands and has an odd number of valencies left, is called a *valency-odd* cycle. This cycle cannot be processed without a further in- or out-edge as an edge always connects two vertices, satisfying two demands. Edges can therefore only satisfy an even number of vertices. From now on we will consider tunnels starting with the innermost cycle and working towards the outermost cycle, when processing tunnels. Thus, we say that valency-odd cycles hand over unevenness by adding an additional out-edge. The first valency-odd cycle C that is reached needs an additional out-edge. If the outer neighbour of C was no valency-odd cycle before, it now has an odd number of valencies left, because cycle C handed over unevenness to its outer neighbour. Thus, the cycles between two valency-odd cycles need to provide two additional valencies, one to accept the additional in-edge and another to hand over the unevenness to the next cycle. The unevenness will be handed over until the next valency-odd cycle is reached and it becomes a valency-even cycle when it receives the unevenness of its inner neighbour. There must always be a receiving valency-odd cycle, as the overall number of vertices must be even.

Triangle-Square-Triangle Graph: If the tunnel is a 3-4-3 tunnel as in Fig. 4.4, this graph is called a *Triangle-Square-Triangle* (TST) graph. In order to make the graph 3-regular, the innermost triangle needs to connect all its three vertices with three vertices of the 4-cycle. In this case the 4-cycle has only one vertex left but the outermost triangle also needs to connect all its three vertices with three of the 4-cycle. Therefore, this graph is not augmentable because it is not possible to add edges in a planar way such that every vertex has degree 3.

These observations will help finding a planar and 3-regular augmentation for tunnels. We will first consider tunnels, whose middle cycles are even cycles and the outermost and innermost cycles are odd cycles. We denote such tunnels *odd-even-...-even-odd tunnels.*

Lemma 4.1. *Let T be an odd-even-...-even-odd tunnel with an even number of vertices but not a TST graph. Tunnel T then admits a planar, 3-regular and connected augmentation.*

Proof. We first examine tunnels whose outermost or innermost cycle is no triangle. As every middle cycle needs two in- and out-edges and the innermost and outermost cycle needs one in- or out-edge, we note that this graph has no valency-odd cycle. We process all middle cycles with Rule 3.1 except two vertices. This is possible because all middle cycles are even cycles. The outermost and innermost cycles are augmented according to Lemma 3.3 such that all vertices except one have degree 3. This is possible because the outermost and innermost cycle are odd cycles with at least five vertices. When applying the rules it is important to pay attention what to do with the inner neighbours when adding edges. A cycle C that provides a valency for the inner neighbour has to be augmented in such a way that this valency is incident to a face that is also incident to the valency that the inner neighbour provides cycle C.

Now every middle cycle is planar and 3-regular except two vertices. The outermost and inner cycle is planar and 3-regular except one vertex. Thus, we connect all remaining

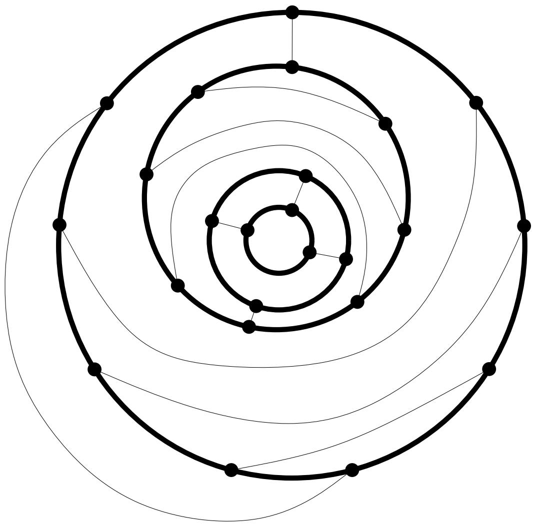


Figure 4.5.: A connected augmentation of a 3-4-8-9 tunnel.

valencies in such a way that every middle cycle has one in- and one out-edge. The resulting graph is still planar, 3-regular and additionally connected.

In case the outermost or innermost cycle is a triangle, it cannot be processed by Lemma 3.3. In this case all vertices must be connected with the inner or outer neighbour. As this neighbour must have at least four vertices it still can provide one valency to connect the graph. Thus, a graph that has a triangle as a outermost and innermost cycle can also become planar, 3-regular and connected. □

Figure 4.5 shows such an augmentation for a 3-4-8-9 tunnel. Lemma 4.1 shows that an augmentation is possible if the outermost and innermost components are odd cycles and all middle cycles are even cycles. We added a minimum number of in-edges and out-edges such that the graph is connected. All other vertices of each cycle gain degree 3 by connecting them with another vertex of the same cycle which is possible because all of these cycles are valency-even cycles. We now examine the case when a graph also contains valency-odd cycles.

Theorem 4.2. *A tunnel with an even number of vertices admits a planar, 3-regular and connected augmentation if and only if (1) the tunnel is a 3-3 tunnel or (2) does not start or end with two triangles or (3) is no TST graph.*

Proof. An augmentation for a tunnel can be found by starting with the innermost cycle and working towards the outermost cycle. The cycles are processed like they are in Lemma 4.1 until the first valency-odd cycle C is reached (meaning the first odd middle cycle or an even innermost cycle). The cycle C needs two out-edges as it has to hand over unevenness to its outer neighbour. Thus we do not apply Rule 3.1 except for two vertices but Rule 3.1 except for three vertices. Now one in-edge and two out-edges can be added. The outer neighbour of cycle C has to accept two in-edges and has to add at least one out-edge. As a cycle has at least three vertices, the outer neighbour is always able to accept two in-edges and add one or two out-edges by applying Rule 3.1 except for three or four vertices.

The only case when a cycle has to accept more than two in-edges (namely three in-edges) occurs, when the innermost cycle is a triangle. In this case the outer neighbour needs to accept three in-edges and has to add at least one out-edge requiring a minimum of four vertices. Thus, if the outer neighbour of an innermost triangle is also a triangle, the graph is not processable except the graph is a 3-3 tunnel. □

4.3. Augmentation of Tunnels Without Regarding Connectivity

With Theorem 4.2 we can find a solution how to augment tunnels such that they become planar, 3-regular and connected. This is only possible for the 3-3 tunnel or if the tunnel does not start or end with two triangles and if it is no TST graph. We will show that we can process tunnels that start or end with two triangles if we are not regarding connectivity. We denote a tunnel that consists of an even number of triangles, followed by a TST and closes with an even number of triangles a *TST-containing tunnel.* TST-containing tunnels are for example the TST tunnel itself, the 3-3-$\underbrace{3\text{-}4\text{-}3}_{TST}$-3-3 tunnel or the 3-3-3-3-$\underbrace{3\text{-}4\text{-}3}_{TST}$ tunnel.

Theorem 4.3. *A tunnel $T = (V, E)$ with an even number of vertices admits a planar and 3-regular augmentation if and only if it is no TST-containing tunnel.*

Proof. If the graph is a 3-3 tunnel or is not starting or ending with two triangles, we use Theorem 4.2 to find an augmentation for T. Otherwise, if the graph starts or ends with two triangles *abc* and *a'b'c'* we connect the triangles by adding the edges $\{a, a'\}$, $\{b, b'\}$ and $\{c, c'\}$. These two triangles are now 3-regular and need not to be regarded anymore. If the tunnel T minus *abc* and *a'b'c'* still starts or ends with two triangles, those two triangles are satisfied in the same way.

This elimination of triangle pairs is repeated until either the whole graph is fully processed or until there is a subgraph T' which does not start or end with two triangles that is not 3-regular yet. As T' is no TST tunnel, as T is not a TST-containing tunnel, we can process T' by using Theorem 4.2.

Conversely, a graph is no TST-containing tunnel if it admits a planar and 3-regular augmentation, as a TST tunnel cannot be augmented under any circumstances. If a graph is a TST-containing tunnel, two triangles that are at the end or the start of a tunnel can only be processed by adding three edges and connecting these triangles. They can be processed and are isolated from the rest of a tunnel and there would be only a TST graph left, which allows no augmentation. Thus, if the tunnel admits a planar and 3-regular augmentation, it is not a TST-containing tunnel. □

4.4. 2-Connected Augmentation of Tunnels

After having examined a connected, planar and 3-regular augmentation, we now want to find an augmentation that is not only connected but 2-connected. In order to make a tunnel 2-connected, every middle cycle needs at least two in-edges and two out-edges because of the lower bound of edges for 2-connectivity. Thus, all middle cycles need at least four vertices. The outermost and innermost cycle need two in- or out-edges. Valency-odd cycles need to hand over uneveness by adding an additional in-edge until another valency-odd cycle receives unevenness as already observed. Thus, valency-odd cycles have two in-edges and three out-edges (or vice versa). Valency-even cycles between valency-odd cycles have three in-edges because they are handed over unevenness. After having received a third in-edge they have an odd number of valencies left and therefore they must add three out-edges. Thus, cycles between valency-odd cycles need at least six vertices. We first examine what kind of cycles need to accept two in-edges and what kind of cycles need to accept three in-edges.

Lemma 4.4. *A middle component C of a tunnel needs to accept three in-edges from its inner neighbour if the number of vertices inside C is odd. Otherwise, if the number of vertices inside a middle component is even, the middle component needs to accept two in-edges from its inner neighbour.*

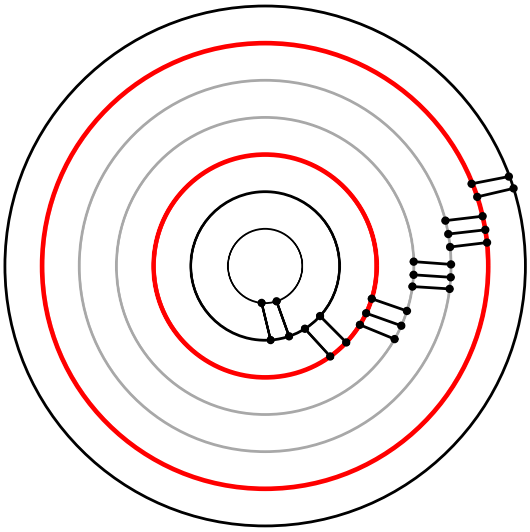


Figure 4.6.: A schematisation of a 2-connected tunnel. Valency-odd cycles are marked red and cycles between two valency-odd cycles are marked grey.

Proof. This lemma is a continuation of the observation of handing over unevenness. Every middle cycle needs two in-edges and two out-edges in order to be 2-connected. The innermost and outermost cycles need two in- or out-edges. Thus, a tunnel that only consists of even cycles only consists of valency-even cycles. A valency-odd cycle must be therefore an odd cycle. If a cycles C gets three in-edges, it must be because of unevenness that was handed over from a valency-odd cycle. Therefore all cycles processed before, all cycles inside cycle C, must have an odd number of vertices. □

We can therefore show the following theorem.

Theorem 4.5. *A tunnel with an even number of vertices admits a planar, 3-regular and 2-connected augmentation if and only if (1) the tunnel has no triangle as a middle cycle and (2) all cycles that have an odd number of vertices inside consist of at least five vertices.*

Proof. We know that a cycle lies between two valency-odd cycles if the number of vertices inside is odd. Thus, we can process the tunnel as follows. We apply Rule 3.1 except for four vertices on valency-even middle cycles that are not between valency-odd cycles (meaning even cycles with an even number of vertices inside). On all valency-even middle cycles between valency-odd cycles (meaning even cycles with an odd number of vertices inside), we apply Rule 3.1 except for six vertices.

All remaining middle cycles that were not yet augmented are valency-odd cycles and therefore augmented by Rule 3.1 except for five vertices. Now only the innermost and outermost cycle have to be augmented by Rule 3.1 except for two or three vertices, dependent on whether they are even or odd cycles.

Now every cycle has enough valencies left to add the lower bound of edges needed to achieve 2-connectivity. There are also enough valencies left to hand over unevenness if needed, as we kept six valencies of each cycle that lies between two valency-odd cycles. Thus, all cycles can be connected by connecting all remaining free valencies with in- and out-edges.

A cycle that is between two valency-odd cycles gets three in-edges and needs at least two out-edges. Hence, four vertices are not sufficient and the cycle must have at least five vertices. Fig. 4.6 is a schematisation of such an augmentation. □

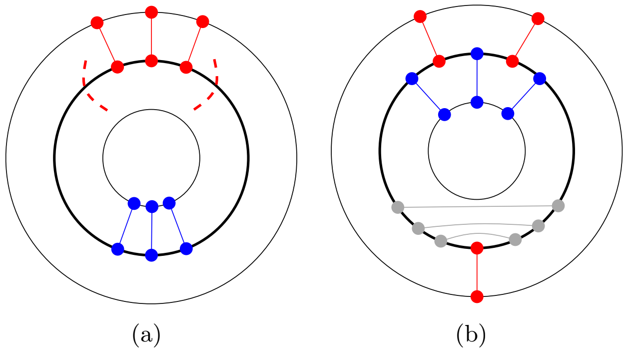


Figure 4.7.: Two tunnels are shown and the middle cycle has three in-edges and three out-edges. In Figure (a) we can find a cut consisting of two edges such that the tunnel is not connected anymore. In Figure (b) the in-edges and out-edges are added alternately and we cannot find a cut consisting of less than three edges. It is shown in grey how the remaining valencies and be satisfied.

4.5. 3-Connected Augmentation of Tunnels

When a 3-connected augmentation is sought, the augmentation is similar to the 2-connected one with the exception that every cycle needs at least three in-edges and three out-edges, valency-odd cycles need three in-edges and four out-edges (or vice versa) and cycles between valency-odd cycles need four in-edges and four out-edges.

Theorem 4.6. *A tunnel with an even number of vertices admits a planar, 3-regular and 3-connected augmentation if and only if (1) all middle cycles consist of at least six vertices and (2) all middle cycles that have an even number of vertices inside consist of at least seven vertices.*

Proof. When seeking a 3-connected augmentation, middle cycles with an odd number of vertices inside, need six or seven valencies (dependent on whether they are even or odd cycles). Middle cycles with an even number of vertices inside need eight valencies. The outermost and innermost cycle need three or four valencies.

When processing the cycles, the choice of the vertices for the in- and out-edges is important. In the 2-connected case, a middle cycle consisting of four vertices can provide two arbitrary valencies for the outer neighbour and two arbitrary valencies for the inner neighbour. But in the 3-connected case, if we choose three vertices that form a path to provide three valencies for the same neighbour, the graph is not 3-connected as it is sufficient to remove two edges in order to separate two cycles as shown in Fig. 4.7. Instead all in- and out-edges have to be assigned to the free valencies alternately as in Fig. 4.7. It is clear that we cannot use Rule 3.1 to process a tunnel. Instead we alter Rule 3.1: If we need k valencies of a cycle C we choose one arbitrary vertex v. One of the out-edges will be assigned to this vertex v. Then we choose $k-1$ vertices that have the largest distance to v. As already described, the needed in- and out-edges will be assigned alternately on the valencies. All other remaining vertices are augmented by edges in the inner face. □

5. Planar 3-Regular Augmentation of 2-Regular Graphs with Fixed Embedding

After having examined tunnels, we now examine augmentations for general 2-regular graph with fixed embedding.

5.1. 2-Connected FEPRA-2G

In Theorem 4.5 it was shown that every tunnel admits a planar, 3-regular and 2-connected augmentation if no middle cycle is a triangle, the graph has an even number of vertices and every cycle that has an odd number of vertices inside has at least five vertices. We can show that any 2-regular graphs with these conditions can be augmented such that it becomes planar, 3-regular and 2-connected. We first show how we can augment all cycles that are incident to a common face. Later we use this to augment the whole graph.

Let $G = (V, E)$ be a graph that fulfills these conditions and T its tree representation. Let a cycle C, represented by the node N, have k children. All k children are augmented in such a way that each of the k children $K_1, K_2, \ldots, K_k$ and their subtrees are planar. Each child is additionally 3-regular except for two or three vertices of K_i and 2-connected.

Lemma 5.1. *The cycle C and its subtree can be augmented such that it becomes planar, 2-connected and all vertices have degree 3 except for two or three vertices of C that have degree 2.*

Proof. If cycle C and its k subtrees have an even number of vertices, we omit two vertices of C. These omitted two valencies are the valencies of C that we are demaning. As C consists of at least four vertices as demanded, C still has two valencies and we can apply Rule 3.1 except for two or three vertices on cycle C if there are still more than three vertices left. Now all components, C and its child-subtrees are planar connected components where each component is 2-connected and has two or three valencies. All these valencies are incident to the inner face of cycle C. Now, all components can be augmented such that C and its subtree become planar, 3-regular (except for two vertices that we have omitted before processing C) and 2-connected using Lemma 3.5.

If cycle C and its subtrees have an odd number of vertices, we omit three vertices. Cycle C has at least five vertices in this case. Thus, there are still at least two vertices left and we can augment C and the other components as described. Figure 5.1 shows an example. □

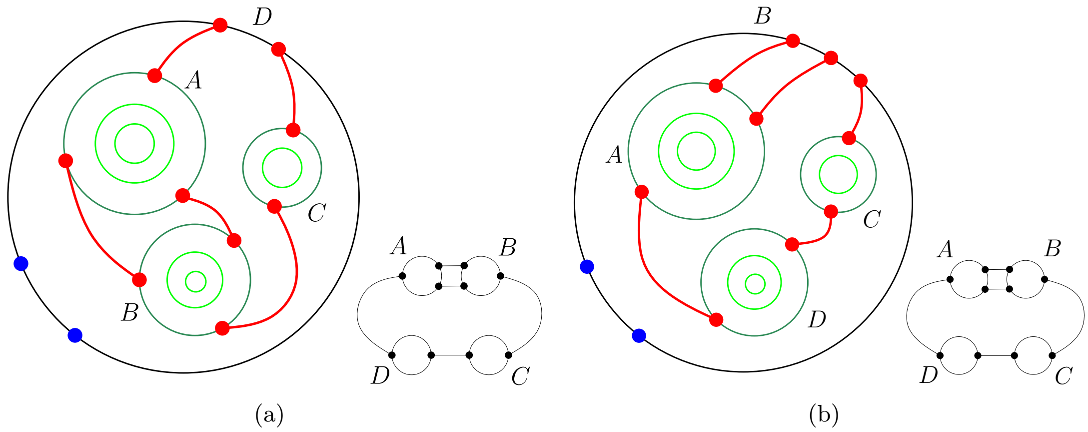


Figure 5.1.: Figure (a) shows a planar, 3-regular and 2-connected augmentation of a subgraph with three components in a cycle. All vertices of a cycle have degree 3 except for two or three vertices marked in red. The sum of the valencies is even. Figure (b) shows such an augmentation if the sum of valencies inside is odd. Next to each augmentation is an augmentation of a graph with corresponding cycles according to Theorem 3.6.

Theorem 5.2. *A 2-regular graph $G = (V, E)$ with an even number of vertices and a fixed embedding admits a planar, 3-regular and 2-connected augmentation if and only if (1) the graph has no triangle as a middle cycle and (2) all cycles that have an odd number of vertices inside consist of at least five vertices.*

Proof. Let tree T be the representation of G and let h be the depth of the tree T. We can augment the graph by starting processing the cycles with depth 0 and progress by climbing up the tree.

If the current node is a leaf, we use Rule 3.1 except for two or three vertices.

If the current node N is not a leaf, it has k children. These children all have two or three valencies left. Then node N and its subtree can be augmented using Lemma 5.1. After the augmentation node N and all its children are planar, 2-connected and all vertices have degree 3 except for two or three vertices with degree 2.

This way all cycles can be processed until we reach the nodes with depth $h - 1$. If there is only one of such node, it must have two valencies left, because the whole graph has an even number of vertices and all other vertices have degree 3. Then, the last two valencies can be satisfied by one edge in the outer face and the whole graph is then planar, 3-regular and 2-connected.

Else, if there are multiple nodes with depth $h - 1$, it means that several components with two or three valencies are drawn next to each other as in Fig. 5.2. The components can then be augmented by using Lemma 3.5.

Conversely, if a graph admits a planar, 3-regular and 2-connected augmentation, it has no triangle as a middle cycle, has an even number of vertices and all middle cycles with and odd number of vertices inside consist of at least five vertices.

If the graph had a triangle as a middle cycle, it would be impossible to connect the triangle to its inner and outer neighbour with two edges each, because it only provides three valencies. But if the triangle is connected with a neighbour with only one edge, the edge is a bridge and the graph is not be 2-connected.

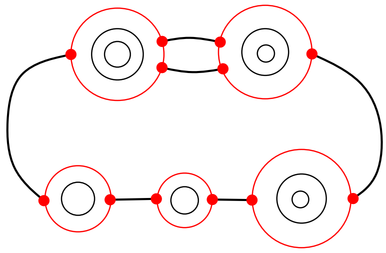


Figure 5.2.: This schematization shows five components that are processed such that they are planar, 3-regular and 2-connected except for two or three vertices of the outermost cycles, here marked in red. These can be augmented such that the whole graph becomes planar, 3-regular and 2-connected by using Lemma 3.5.

If a middle cycle has an odd number of vertices inside, it means that there is an odd number of valency-odd cycles inside. That means that the middle cycle will get unevenness handed over and therefore three in-edges. In order for the graph to be 2-connected, the cycle needs at least two out-edges. Therefore the middle cycle must have at least five vertices. □

5.2. 3-Connected FEPRA-2G

The difference between a tunnel and a general 2-regular graph is that, inside a cycle, there can be multiple components. A tunnel admits a planar, 3-regular and 3-connected augmentations if the tunnel has middle cycles with at least six vertices, every middle cycle that has an even number of vertices inside has at least seven vertices and the tunnel has an even number of vertices.

We now consider a graph $G = (V, E)$ that fulfills these conditions and its tree representation T. Let a cycle C, represented by the node N, have k children. All k children are augmented in such a way that each of the k children $K_1, K_2, \ldots, K_k$ and their subtrees are planar and all vertices have degree 3 except for three, four or five vertices of each K_i that have degree 2. For all vertices u, v with degree 3 it is $\lambda(u, v) = 3$ and for all vertices s, t with s having degree 2 it is $\lambda(s, t) = 2$.

In the previous section we showed by Lemma 3.7 how to augment a graph that consists of connected components that are 2-connected and have four or five valencies left. Lemma 3.11 shows how to augment triangles and Theorem 3.11 shows how to augment a graph with triangles and cycles with four or five valencies such that the graph becomes planar, 3-regular and 3-connected. We can show by these lemmata and by the theorem, that we can also augment C and its children. Later we will use this to augment a whole graph by processing the nodes of its tree representation bottom-up.

Lemma 5.3. *There exists an augmentation of C and its subtree such that it becomes planar and 3-connected. All vertices of the resulting subgraph have degree 3 except for four or five vertices of C that have degree 2.*

Proof. We will construct a graph H with the same number of valencies as C and its children. We will choose H in such a way that we know a planar, 3-regular and 3-connected augmentation of H and show that the augmentation for H corresponds with a planar, 3-regular and 3-connected augmentation for C and its children.

Let x be the number of triangles, y the number of components with four valencies and z the number of components with five valencies inside cycle C.

If the number of vertices in the subtree of C is odd, we apply Rule 3.1 except for six or seven vertices on cycle C. We mark the valencies of C alternately. Every second vertex is

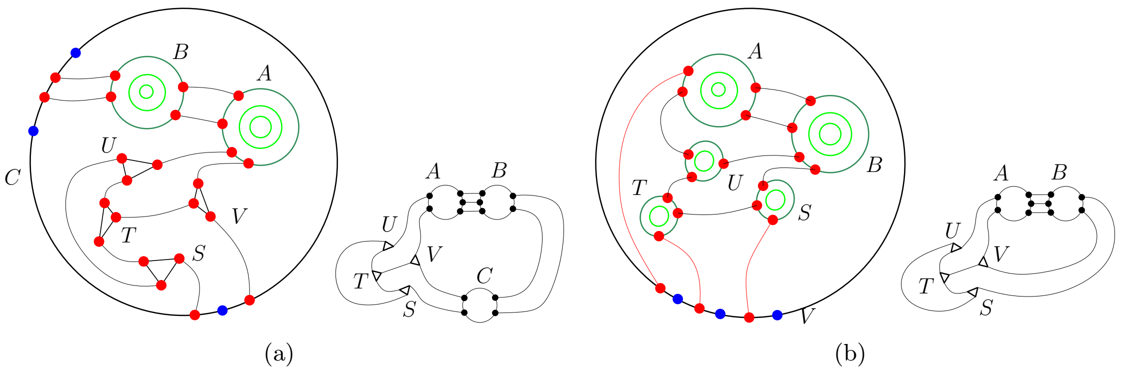


Figure 5.3.: Figure (a) shows a planar, 3-regular and 3-connected augmentation of a subgraph with multiple components in a cycle. All vertices of a cycle have degree 3 except for three or four vertices marked in red. The sum of the valencies is even. Figure (b) shows such an augmentation if the sum of valencies inside is odd. Next to each augmentation is an augmentation of a graph with corresponding cycles according to Theorem 3.11.

reserved for in-edges (but at most three valencies). All remaining vertices are reserved for out-edges.

We then consider a graph H consisting of $x+1$ triangles, y even cycles and z odd cycles. Then, we apply Theorem 3.11 on H and find a planar, 3-regular and 3-connected augmentation. As H is a planar graph, we can draw it in such a way that the inner face of the one additional triangle T becomes the outer face of H. Triangle T corresponds to the outer cycle C, an even cycle corresponds to a component with four valencies and an odd cycle is a corresponds to a component with five valencies.

We now could substitute the components of the augmented graph H with corresponding children of C and C itself. The augmentation edges of H are also planar, 3-regular and 3-connected augmentation edges of C and its children. We use the augmentation edges of H to augment C and its children.

If the number of vertices in the subtree of C is even, we construct the following. We apply Rule 3.1 except for seven or eight vertices on cycle C. We mark the valencies of C alternately. Every second vertex is reserved for in-edges (but at most four valencies). All remaining vertices are reserved for out-edges. We then consider a graph H consisting of x triangles, $y+1$ even cycles and z odd cycles and augment it using Theorem 3.11. Now the additional even cycle E is pulled over all other cycles and stands as a substitute for cycle C. Graph H equals an augmentation for G.

As G is a subdivision of graph H, G is also 3-connected except for the four of five valencies of C. □

Theorem 5.4. *A graph with an even number of vertices admits a planar, 3-regular and 3-connected augmentation if and only if (1) all middle cycles consist of at least six vertices and (2) all middle cycles that have an even number of vertices inside consist of at least seven vertices.*

Proof. We examine the graph bottom up, by starting at the leafs of the three representation and working towards the root. We can show that each cycle fulfills the condition of Lemma 5.3.

Starting with the leafs, we can use Rule 3.1 except for four or five vertices, if the cycle consists of more than three vertices. If not, we just leave the triangle as it is. Now, the next cycle only has components with either three, four of five valencies inside. It can therefore be augmented such that it becomes planar, 3-regular and 3-connected except for three or four vertices using Lemma 5.3.

Thus, all cycles fufil the conditions and all subtrees can be augmented such that they are planar, 3-regular and 3-connected except for three of four vertices. If we have reached the root there are two possibilities. Either there is only one cycle enclosing all other cycles. Then it must have four valencies left, because all vertices have degree 3 and the number of vertices is even. Thus, the number of vertices with degree 3 must be even as well. These four valencies can be satisfied by two edges running through the outer face. Else if there are multiple cycles in the outer face, Lemma 3.7 can be used to make the whole graph planar, 3-regular and 3-connected.

Figure 5.3 shows examples of such an augmentation. □

5.3. Connected FEPRA-2G

We have seen that the conditions for a tunnel that admits a planar, 3-regular and 2-connected or 3-connected augmentation are the same as the conditions for a graph that admits a planar, 3-regular and 2-connected or 3-connected augmentation. In case of a connected augmentation, the conditions have still to be examined. We observe that a minimum of valencies is necessary to connect components that are incident to a common face. Therefore, it is difficult to describe a concrete family of graphs that admit a solution for connected FEPRA-2G. But we can present an algorithm which is able to decide whether the graph admits a planar, 3-regular and connected augmentation and find such an augmentation.

We consider a graph $G = (V, E)$ and its tree representation T. Let cycle C, represented by node N, have $k-1$ children. All $k-1$ children are augmented in such a way that each of the $k-1$ children $K_1, K_1, \ldots, K_{k-1}$ and their subtrees are planar and connected. All vertices of K_i and its subtree have degree 3 except for t vertices of K_i that have degree 2 for $i = 1, 2, \ldots, (k-1)$. We note that C and its children are incident to a common face f. We will now focus on C and its children K_i and refer to them as components.

The number of components that can provide one valency to the face f is r. We call such a component *donator* as it can provide a valency to any other component. The number of components that can provide at least two valencies to the face f is s. We call such a component a *chain link* because we will show in the following that we use such components to create a chain.

Lemma 5.5. *Cycle C and its subtree can be augmented such that it becomes planar, 3-regular and connected if (1) the number of valencies that can be provided to the common face f is even and (2) the number of valencies is at least $2k-2$ and (3) every component provides at least one valency and (4) neither C nor one of its children is a triangle. Cycle C provides as few valencies as possible to fulfil these conditions.*

Proof. We will first show that it is possible to add edges such that all the components are connected. After having ensured that we can connect all components, we will explain in detail which valencies are connected with each other in order to achieve a 3-regular graph.

As every component provides at least one valency, it is $k = r + s$. All components provide $2k - 2 = 2(r + s) - 2$ valencies in total. As there are $2(r + s) - 2 > r$ valencies, there must be at least one chain link and thus $s > 0$. We connect all chain links such that the

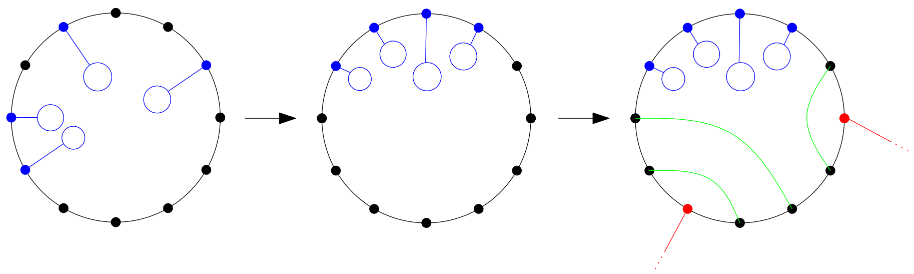


Figure 5.4.: This figure shows how vertices of a cycle that are connected to inner neighbours are moved together. Then, two vertices with the largest distance to each other are chosen, allowing all other valencies to be satisfied by adding edges connecting the with each other.

components form a chain P by adding edges. Now $2s - 2$ valencies are gone and there are still at least $2r$ valencies left in total. All r donators are not connected yet. The r donators have r valencies, thus the chain must provide at least r valencies. We can now connect all remaining donators to the chain P. That way, all components are connected with each otheforming a tree T' of components.

Now we have shown that it is possible to make all components connected if there are $2k - 2$ valencies available. We will now go in further detail how we can find a planar, 3-regular and connected augmentation. If there are exactly $2k - 2$ valencies, the graph would be already 3-regular after connecting the components. But if there are more than $2k-2$ valencies, it is possible that after creating the tree T' there are still chain links that have valencies left. These valencies have to be satisfied. As C provides as few valencies as possible, all of the valencies of C are needed for the tree and C has not got any valencies left after connected the components to the tree T'. This does not apply to the children. If a component has an even number of valencies left, we want to use Rule 3.1 to connect all remaining valencies. We also want to use Rule 3.1 on all other components that have an odd number of valencies left such that they have only one valency left. We will then use depth-first-search to satisfy all valencies that are remaining after using Rule 3.1.

Thus we will find a planar, 3-regular and connecting augmentation in three steps. We will create T' such that all components are connected. Afterwards, we will use Rule 3.1 to satisfy all components with an even number of valencies after having created T'. At last we will use Rule 3.1 and depth-first-search to fully process C and its children. In order to use Rule 3.1 which is actually used on empty cycles on our components, we have to choose the valencies carefully when connecting the components to T'.

We will now explain in more detail, which valencies we choose when connecting the components. A component has t valencies and q vertices that have already degree 3 (for in-edges to connect the component to inner neighbours). If a component has q vertices with degree 3, we will re-sort these vertices by moving all these vertices together such that they form a path as in Fig. 5.4. This is possible, because all these in-edges must be connected to an inner neighbour and thus we will not create a double edge which is not allowed.

Now, we can deal with chain links as if they were t-cycles. If a component is in the middle of the chain P it needs two valencies u and v to construct the chain link. These valencies u and v are chosen in such a way that they have the largest distance to each other, thus leaving all remaining valencies forming two paths P_1 and P_2 of the same length (or differ by one if there are odd valencies left) when ignoring the q vertices with degree 3 (see Fig. 5.4). If a component is at the start or the end of the chain P, it only needs one valency u for the tree. We now have to choose this valency in such a way that the q

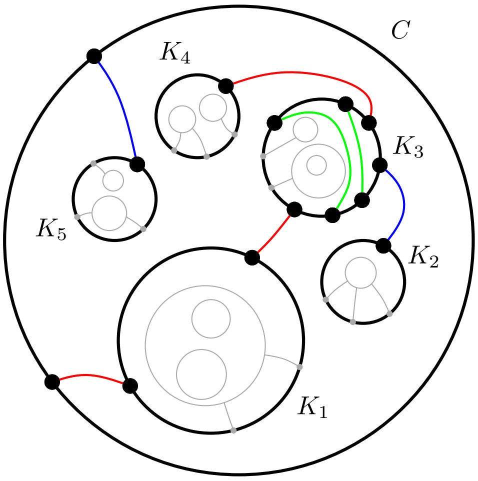


Figure 5.5.: An example of an augmentation using Lemma 5.5. Red edges are the edges to create the chain P. Donators are connected to the chain with blue edges. The remaining valencies are satisfied with green edges.

vertices with degree 3 and the one valency used for the chain P split the other valencies into two paths of the same length (or differ by one). We connected all chain links to a chain. Now, we have to append the donators to the chain. When appending the donators, we will connect them with valencies that are adjacent to u or v such that the paths of the remaining valencies P_1 and P_2 are still either of the same length or differ by at most one. Now we have constructed the tree T'.

After having connected all components to a tree T', we can apply Rule 3.1 except for the vertices with degree 3 on components with an even number of vertices even though they are not cycles. This happens by connecting all pairs of valencies with the same distance to vertex u while ignoring the q vertices.

If a chain link has an odd number of valencies left, we can also apply Rule 3.1 on all of them, but there will be one valency of each component left that cannot be satisfied. But now all components have at most one valency remaining. We choose an arbitrary component as the root of T'. With a depth-first-search in T' we look for components with a valency and connect the vertex with the next valency that we can find. As T' is a tree, we can embed the edge parallel to the path from one valency to the next valency found in T' which ensures that the edges are added in a planar way. □

Figure 5.5 shows an example how to augment components with Lemma 5.5. We will now extend the lemma and examine augmentations when there are also triangles.

Lemma 5.6. *Cycle C and its subtree can be augmented such that it becomes planar, 3-regular and connected if (1) the number of valencies that can be provided to the common face f is even and (2) the number of valencies is at least $2k-2$ and (3) every component provides at least one valency and (4) the set of components does not consist of just a triangle and a donator. As C provides as few valencies as possible to fulfil these conditions and the set of components must not consist of just triangle and a donator this means in return that C must not be donator if the only child is a triangle.*

Proof. If none of the components is a triangle, we use Lemma 5.5. If we would use Lemma 5.5 even though there are triangles in the graph, we would still be able to construct the tree T' but in the second step it is possible that we cannot use Rule 3.1 on all

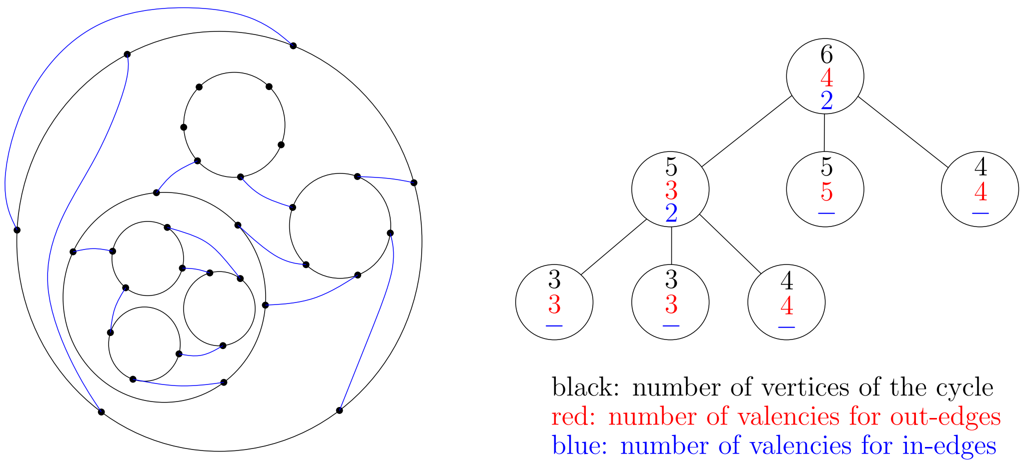


Figure 5.6.: A 2-regular graph with a possible connected augmentation. On the right side is the tree representation of the graph.

components that have an even number of valencies left. If there is a triangle that has two valencies left we cannot use Rule 3.1 on the remaining valencies because they are already connected and this would result in a double edge. We denote such triangles by *trouble triangles*.

A trouble triangle occurs, if a triangle is the start or the end of the chain P. We will therefore try to create the chain P in such a way, that no triangle marks the start or the end of the chain P. The graph can then be processed by Lemma 5.5. This is possible unless all chain links are triangles or all chain links except one are triangles. In case all components are triangles, they can be easily processed by using Lemma 3.9.

Otherwise if not all of the components are triangles and there is either one or two trouble triangles there are two possibilities. Either there are at least as many donators as there are trouble triangles or not. If there are as many donators as trouble triangles, we can connect the donators with the trouble triangles, such that they provide less than two valencies and we can again process the components by Lemma 5.5.

Else if there are not as many donators as there are trouble triangles, we still connect the donator (if available) with one trouble triangle which means that there is always only one trouble triangle left (if all components are triangles, they can be processed, if all chain links were triangles but there are donators, there must be at least one donator leaving only one trouble triangle or there is one chain link that is no triangle such that there is only one trouble triangle). We choose this trouble triangle as the root and do a depth-first-search in P. We connect one valency of the triangle with the first valency that we find. As the chain is a path of components, we can connect them in a planar way. Afterwards all components are connected and there exists no trouble triangle. We can then satisfy all remaining valencies as by using step 3 of Lemma 5.5.

The only case when we cannot find another valency through depth-first-search occurs, when C has only one child K_1 and they are one triangle and a donator. □

With this lemma, it is shown that a solution for a connected FEPRA-2G is dependent on the number of valencies that can be provided for a face. We solve connected FEPRA-2G for a graph by examining it locally and we did not find a simple description of all 2-regular graphs for which connected FEPRA-2G is solvable.

Knowing that each face with k incident components needs at least $2k - 2$ valencies that are assigned to the face, we claim that the procedure described in Algorithm 5.1 computes whether there is an augmentation for a 2-regular graph $G = (V, E)$ such that it becomes planar, 3-regular and connected.

We know that k components that are incident to a common face can be connected if there are $2k - 2$ valencies and each component provides at least one valency. A 2-regular graph consists of cycles and cycles are always incident to exactly two faces. Therefore, we can start with examining the outer faces of the cycles, that are the leafs with the lowest depth in the tree representation. We will try to connect all components incident to the face with as few valencies of the outer cycle as possible and preserving as many valencies for the next outer face as possible. After that we will examine the next outer face, trying to connect the components with as few edges as possible, processing the cycles by climbing towards the root in the tree representation. Therefore, we can check whether a graph admits a planar, 3-regular, connected augmentation by the greedy Algorithm 5.1.

Algorithm 5.1: ConnectedFepra-2G

```
Input: A tree representation T = (N, L) of a 2-regular graph G = (V, E) with m
       connected components
Output: true if G has a solution for connected FEPRA-2G, false if not
N = (c_1, c_2, ..., c_m) is a sorted list with depth(c_i) ≤ depth(c_{i+1});
for i = 1, 2, ..., n do
    c ← c_i;
    if c is a leaf then
        outEdges ← c.numberOfVertices;
    else
        c.valencies ← 0;
        for child ∈ c.children do
            c.valencies ← c.valencies + child.outEdges;
        end
        c.inEdges ← max{1, 2 · c.numberOfChildren − c.valencies};
        if (c.valencies + c.inEdges) is odd then
            c.inEdges ← c.inEdges + 1;
        end
        if c.numberOfChildren = 1 and c.children is a triangle then
            c.inEdges ← c.inEdges + 2;
        end
        c.outEdges ← c.numberOfVertices − c.inEdges;
    end
    if (c.outEdges < 0)
    ∨ (c is a middle cycle or one of many outermost cycles ∧ c.outEdges < 1)
    ∨ (c is the only outermost cycle ∧ c is a triangle ∧ c has only one inner
    neighbour which has only one valency) then return false;
end
return true;
```

A leaf in the tree representation will provide all its valencies to its outer face, therefore it provides as many vertices as it consists of. For every other cycle C, we need to compute how many vertices it has to use for its inner face f and how many it can provide its outer face. First, we count how many valencies are already provided by the $k - 1$ children of C. If the sum of valencies is already at least $2k - 2$, then cycle C needs to provide either one, two or three valencies, because C has to provide at least one valency and C might have to provide two valencies in order to have an even number of vertices for f. In case there is only one child which is a triangle we have to provide even three valencies, else the set of

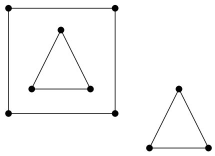


Figure 5.7.: This graph is a TST graph.

components that are incident to that face would be a triangle and a donator which cannot be processed. Otherwise, if the number of valencies inside is less than $2k-2$, cycle C has to provide as many vertices necessary to reach $2k-2$.

A graph admits no planar, 3-regular and connected augmentation, if a middle cycle cannot provide any valencies for its outer face (because it needs too many for its inner face). It also admits no augmentation if an outermost cycle cannot provide enough valencies for its inner face. At last, we have to check whether the outermost component is a triangle with only one child which is a donator. Figure 5.6 shows an example.

If Algorithm 5.1 returns *true*, it computes at the same time a matching of valencies to faces, such that at least $2k-2$ vertices are assigned to each face (except for the innermost faces and the outer face) where k is the number of components incident to a face. With this assignment, we can process a graph $G=(V,E)$ with its tree representation T and depth h, starting with the nodes of depth $h-1$ and working towards the root by using Lemma 5.6.

5.4. FEPRA-2G Without Regarding Connectivity

Before examining FEPRA-2G without regarding connectivity, we first point out that given a 2-regular and planar graph G, we can find an augmentation which is also a valid augmentation for the same graph G with another embedding. This is only possible if we do not really change the embedding but only choose another face as the outer face. This is easy to understand when imagining the graph drawn on a sphere of a ball. Then, we choose an arbitrary face f and roll out the sphere to a plane by splitting the sphere inside the face such that f becomes the outer face. By doing this, it is obvious that a planar and 3-regular augmentation for a graph with a certain embedding is also a planar and 3-regular augmentation for a graph with the same embedding but with another outer face. Thus, when seeking an augmentation for graph G choosing a suitable face is feasible. From now on we will always choose a graph whose outer face is only incident to one connected component. Such a face can always be found by splitting the sphere inside an innermost cycle. This means that the graph shown in Fig. 5.7 is also a TST graph. Thus, starting from now we will not further regard graphs that have multiple components incident to the outer face.

Solving FEPRA-2G without regarding connectivity is easy if there is no triangle in the graph. We will show that in fact any graph can be augmented as long as there is an even number of vertices (and no triangle). We will augment such a graph bottom-up.

Lemma 5.7. *A 2-regular graph G with an even number of vertices admits a planar and 3-regular augmentation if the graph contains no triangle.*

Proof. As in the previous section, we will start with the cycles whose vertices have depth 0 in the tree representation. If the cycle is an even cycle, we use Lemma 3.2 to process the whole cycle. Otherwise, if the cycle is an odd cycle, we use Lemma 3.3 to process the cycle except for one vertex. Lemma 3.3 can be applied, because all cycles consist of at least four vertices.

After we used Lemma 3.2 or Lemma 3.3 there are some cycles with depth 0 that have one valency left. Let C be such a cycle. If C is incident to another component that also has only one valency left, we connect them by an edge. Inside cycles with depth 1 is at most one cycle with a valency. We connect this valency to its parent node.

After that, we process all cycles whose node in the tree representation have a higher depth. If the cycle has an even number of valencies, we again use Lemma 3.2, otherwise Lemma 3.3 and connect two cycles that have one valency and are incident to a common face. Both lemmata are always applicable because a cycle consists of at least four vertices and has at least three valencies because we use at most one valency for the inner neighbour.

These actions are repeated for every cycle until the most outer cycle is reached and the whole graph is processed. As the number of vertices is even, the outermost graph is fully processed as it must have a even number of valencies when beginning to augment it. □

This lemma shows that every graph that contains no triangle can be augmented. When examining general graphs, we have to observe triangles carefully. We also know from Theorem 4.3 that TST-containing tunnels are the only tunnels that cannot be augmented, meaning that a tunnel requires a TST as a subgraph for not being augmentable.

We will therefore first figure out when the process described in Lemma 5.7 does not work when considering triangles.

Triangles with three valencies: When a triangle is going to be processed we cannot use Lemma 3.3 as we would do with other odd cycles that are not triangles. That means that a triangle has three valencies and needs three other valencies in order to become 3-regular. Thus, other cycles that are incident to a common face need to provide three valencies. As compared to Lemma 5.7 it is not sufficient to only provide one valency. We call such a triangle a 3-triangle.

Triangles with two valencies: If a triangle already used up a valency (for example for an inner neighbour), it has two valencies left. In this case, the triangle needs two valencies of other components to becomes 3-regular. We call such a triangle a 2-triangle.

As seen from the observations, we can still use Lemma 5.7 on graphs with triangles but have to deal with triangles in a special way. If we examine sibling components that are incident to a common face (meaning all components that are incident to a common face except for the outer cycle) and there are components that have one, two or three valencies that cannot be satisfied, we connect pairs of components that have the same number of valencies. Thus we connect all pairs of triangles with three valencies, pairs of triangles with two valencies and components (which is not necessarily a triangle) with one valency.

There are three possible cases for remaining valencies after having paired components with the same amount of valencies.

1. There are three remaining components with valencies. They are one component A with one valency, another 2-triangle B with two valencies and a 3-triangle C with three valencies. We can connect A to C with an edge and we can connect B to C with two edges and all valencies are satisfied.
2. There are two remaining components. The components can provide either one and two valencies or two and three valencies or one and three valencies. We connect the component with fewer valencies to the component that provides more valencies.

These components leave either one valency (if the components provide one and two valencies or two and three valencies) or two valencies (if the components provide one and three valencies) unsatisfied.

3. There is one remaining component. This component leaves $i \leq 3$ valencies unsatisfied. If $i > 1$ then the remaining component must be a triangle.

This shows, that the inner components demand at most three valencies from the outer neighbour. This means that if we use Lemma 5.7 on graphs with triangle, the following invariant is true.

Invariant I: Every cycle has to provide at most three valencies to its inner neighbours.

We note that every cycle is able to provide three valencies to its inner neighbours. Thus, it means that a graph can be processed with Lemma 5.7 even if innermost cycles or middle cycles are triangles.

The only case in which a graph cannot be processed by Lemma 5.7 occurs, when the outermost cycle is a triangle which has two valencies left. We call an outermost triangle with two valencies a *trouble triangle*. Thus, we will now find solutions how to process graphs with trouble triangles.

We recap how we will use Lemma 5.7 when applying it on a graph that contains triangles. We process the cycles bottom up starting with the cycles with the lowest depth as follows.

Case 1: On a cycle consisting of more than four vertices with an even number of valencies, we will use Lemma 3.2 (by possibly ignoring the vertices with degree 3). Such a cycle is fully processed and does not demand a valency from an outer neighbour.

Case 2: On a cycle consisting of more than four vertices with an odd number of valencies, we will use Lemma 3.3. Such a cycle is not fully processed and demands a valency from an outer neighbour.

Case 3: A triangle with $i \leq 3$ valencies cannot be augmented and it demands i valencies from an outer neighbour.

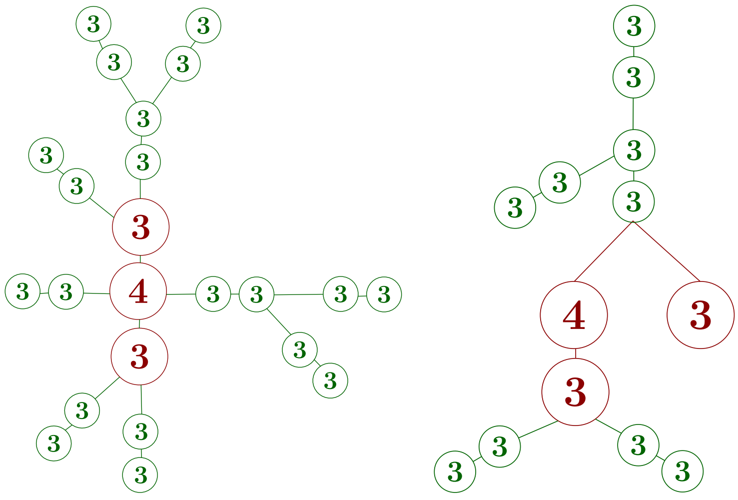
Afterwards, we will connect components that are incident to a common face and have the same depth (and are therefore siblings) by pairing components with the same amount of valencies as already described. After having paired as many components as possible, we have to demand $k \leq 3$ valencies from the parent node.

We will continue processing the whole graph by repeating this procedure on all cycles until the whole graph G is processed. If the outermost cycle is no trouble triangle, then graph G admits a planar and 3-regular augmentation. Otherwise, we will try to modify the augmentation in such a way that we can also satisfy the trouble triangle in the following.

A graph which tree representation is a 3-4-3 path at which we appended 3-3 paths consecutively is called a *TST containing graph* because the core is a 3-4-3 path at which we can append long branches of 3-3 paths.

We can also place a 3-3 path between the 3-4-3 core and split it. Such a graph is also called a TST-containing graph. Figure 5.8 (a) shows an example of a TST-containing graph. A graph that is a 3-4-3 tunnel as in Fig. 5.7 at which we append 3-3 paths is also a TST-containing graph. Figure 5.8 (b) shows an example of such a TST-containing graph.

As we will now try to show how we can modify the augmentation after having used Lemma 5.7 such that a trouble triangle is satisfied we will show the following theorem.



(a) (b)

Figure 5.8.: Example of TST containing graphs.

Theorem 5.8. *Let G be a 2-regular graph with an even number of vertices. Graph G is augmented by Lemma 5.7 and the outermost cycle is a trouble triangle. Graph G admits a planar and 3-regular augmentation if G is not a TST-containing graph.*

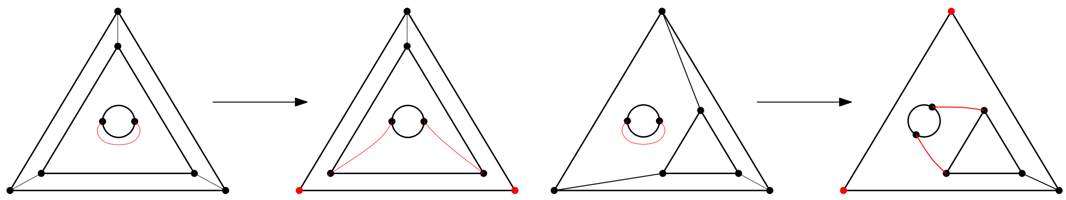
Proof. Let T be the trouble triangle of G. It means that T has an in-edge to an inner neighbour. The inner face if T is the face f. From the observations made, there are only some possible constellation of inner neighbours of T. Either there is only one component that demands a valency which could be a triangle or a bigger cycle or there are two components that demand two and three valencies or one and two valencies. In all cases there can be additionally components that demand zero valencies. We first examine whether we can satisfy the trouble triangle if there are components that demand zero valencies.

If there is a component that is a paired component of two components with the same amount of valencies, we can remove one of those connecting edges. By removing this edge, there are two valencies in f which is why we call it a helping edge. Thus, there are two valencies in f which can be connected to T. Hence, if there is a paired component in the inner face of the triangle, graph G is augmentable.

If there is a component A that demands zero valencies and this component is a cycle that consist of $y > 3$ vertices, it has to possess a handle which is an edge that runs in f and connects two vertices of A. This handle is a helping edge. Hence, if there is a cycle that consists of at least four vertices and does not demand any valencies, graph G is augmentable.

The only component that demands zero valencies and cannot provide two valencies is a component that consists of two triangles that are nested in each other which we call a 3-3 component. This is only true if inside those triangles are not any other components and if there is, it must be another 3-3 component. If there is any component inside of one of those triangles that possesses a helping edge, we can change the augmentation such that f has two valencies as seen in Fig. 5.9.

Now, we examine the actual component with which the trouble triangle is connected. As already mentioned, it can be either a component that demands a valency which could be a triangle or a bigger cycle or there are a 2-triangle and 3-triangle or a 2-triangle and a component with one valency.



(a) (b)

Figure 5.9.: Illustrations how a helping edge inside a 3-3 components can pass valencies to the outer face.

If T is connected to a component which is a 2-triangle thas is paired with a component that demands one valency, there is an edge that connects both valencies. This edge is a helping edge an can be removed in order to satisfy the valencies of the trouble triangle. Hence, if T is connected to a component that is a 2-triangle and a component with one valency, graph G is augmentable.

The same applies to the case if T is connected to a component that consists of a 2-triangle and a 3-triangle. Hence, if T is connected to a component that is a 2-triangle and a 3-triangle, graph G is augmentable.

The last case to consider is when T is either connected with a 1-triangle, a 4-cycle that has one valency or another cycle with more than four vertices that provides a valency. If there is a x-cycle with $x > 4$, then this cycle has to have a handle because it has to provide at most three valencies to its inner neighbours. A 6-cycle that provides three valencies, has three valencies left and adds a handle. The most crucial cases are therefore the case if T is connected to a 1-triangle or to a 4-cycle with one valency.

T is connected to a 4-cycle with one valency: If there is a 4-cycle with only one valency, it means that it provides three of its valencies to an inner neighbour. We know that only a triangle can demand three valencies. Thus, T must be connected to a 4-cycle which has a triangle inside. This is a TST graph and we know that it admits no planar and 3-regular augmentation. We also observed that we can append 3-3 components as we like. A TST graph with appended 3-3 components is a TST-containing graph and it canno be augmented.

T is connected to a 1-triangle: If T is connected to a 1-triangle T', it means that the T' has two in-edges. Thus, T' is either connected to a 2-triangle or a component which resulted from pairing a 3-triangle with a component with one valency. If T' is connected to a triangle T'', then the situation after T'' is equal to the trouble triangle T because it has one in-edge. Thus, after a path of an odd number of triangles, there must either occur a 4-3 component or a x-cycle with $x > 4$. We already examined both cases.

Otherwise, T' is connected to two valencies of a 3-triangle which is connected to a x-cycle that provides one valency. If it is $x > 4$, the component has a handle which we can remove. We also remove the edges that connect T' to the 3-triangle. Then, we connect the valencies of the 3-triangle and the valencies that we achieved after removing the handle. T' has now two valencies. These valencies can be used to satisfy the two valencies of the trouble triangle. But if $x = 4$, then there must be a triangle inside the x-cycle and the x-cycle has no handle. In this case we can only provide two valencies of the x-cycle if we can satisfy two valencies of the inner triangle with other valencies than the valencies of the x-cycle. We have observed that this is not possible if there are only 3-3 components available. If there are only 3-3 components available, graph G is a TST-containing graph as in Fig. 5.8 (b).

We see that we cannot find an augmentation for a TST graph. We can also append an arbitrary number of 3-3 components to a TST graph and the graph will not admit an augmentation either. Thus, a 2-regular graph with an even number of vertices admits a planar and 3-regular augmentation if it is not a TST-containing graph. □

5.5. Time Complexity of FEPRA-2G

All tunnel augmentations can be computed in linear time, as we start with the innermost cycles and satisfy each of its vertices and move on to the outer neighbour until the outermost cycle is satisfied. The augmentation can therefore be computed in linear time.

We consider a graph $G = (V, E)$. There are k vertices that are part of a cycle that has only one or no outer neighbour and there are m vertices that are part of a cycle that has multiple outer neighbours. It is $|V| := n = k + l$.

Theorem 5.9. *The problem 2-connected FEPRA-2G and 3-connected FEPRA-2G can be solved in $O(n)$ time.*

Proof. The problem 2-connected FEPRA-2G as well as the proclem 3-connected FEPRA-2G are solvable by augmenting the graph bottom-up. As long as there is only one outer neighbour, the cycles are processed just like a tunnel. If there is a cycle consisting of a vertices, it can be processed in $O(a)$ time. Thus, all components with only one outer neighbour, can be processed in $O(k)$.

In case a component has multiple outer neighbours, we process this component and its outer neighbours just like solving 2-connected FEPRA-2G or 3-connected FEPRA-2G for the component and its outer neighbour. If the sum of the vertices of these components is b, then this can be computed in $O(b)$ as 2-connected FEPRA-2G and 3-connected FEPRA-2G can be computed in linea time as shown in Theorem 3.13 and Theorem 3.14. Therefore, all components with multiple outer neighbours are processed in $O(l)$. Thus, 2-connected FEPRA-2G and 3-connected FEPRA-2G can be solved in $O(m + l) = O(n)$. □

Theorem 5.10. *Connected FEPRA-2G can be solved in $O(n)$ time.*

Proof. To solve connected FEPRA-2G, we first need a matching of valencies to faces to prove whether a graph admits a planar, 3-regular and connected augmentation. The Algorithm 5.1 that assigns vertices to a face such that the graph admits such an augmentation needs to evaluate all components bottom up to do that. The algorithm has a time complexitiy of $O(n)$.

After finding such an augmentation, the components are processed by using Lemma 5.6. This lemma shows how to augment components that are incident to a common face. Thus, we have to process each face of the graph. Let the sum of vertices of all components that are incident to the same face be a. The lemma simply connects chain links (connected components with more than one valency) to a chain (is done in $O(a)$) and appends donators (connected components with only one valency) to the chain (is done in $O(a)$). Afterwards we use Rule 3.1 on components with an even number of valencies (is done in $O(a)$) and at last a depth-first-search (is done in $O(a)$). Thus, processing all valencies, that are assigned to a face can be computed in linear time. A cycle is incident to two faces. Thus, a cycle is processed twice (as there might be some vertices assigned to its outer or its inner face) and connected FEPRA-2G can therefore be computed in $O(n)$. □

6. Conclusion

In this work we provided solutions for PRA-2G and FEPRA-2G in the field of graph augmentation problems. We have proved that we can find planar and 3-regular augmentations for 2-regular graphs in linear time. Even if we demand 1-connectivity, 2-connectivity or 3-connectivity of the resulting graph, we can find an augmentation in linear time regardless whether the graph has a variable or a fixed embedding.

In most cases, we have found specific characterisations of the graphs that admit a certain augmentation. Merely for graphs with a fixed embedding that admit a planar, 3-regular and connected augmentation, we could not find a characterisation. For this case, we present a linear algorithm that can decide whether an augmentation exists and if so, how to find one.

Future Work. This work leaves open questions that can be researched. Here are three problems that we considered interesting to examine in the future.

Planar and 3-Regular Augmentations of Graphs with Maximum Degree 2: As planar and 3-regular augmentation problems for 2-regular graphs are solvable in linear time, we strongly assume that planar and 3-regular augmentation problems for graphs with maximum degree 2 are solvable in linear time as well. In a graph with maximum degree 2, vertices with degree 0 or degree 1 can be augmented to become a cycle in most cases such that we can use PRA-2G or FEPRA-2G. If we cannot augment those vertices such that they become a cycle, there are only some special cases that still need to be considered.

Restricted Planar and 3-Regular Augmentations: Hartmann, Rollin and Rutter work on finding a planar and 3-regular augmentation W for a graph $G = (V, E)$ implicates that it is also NP-hard to find an augmentation W if there is a given set $W' \subseteq E^c$ and it is demanded that $W \subseteq W'$. Thus, an interesting question would be whether PRA-2G and FEPRA-2G are solvable in linear time if the augmentation W must be a subset of a given $W' \subseteq E^c$.

Planar and 4-Regular Augmentations of 3-Regular Graphs: This work also leaves the question open how to augment planar and 3-regular graphs such that they become planar and 4-regular. The problem of finding a planar and 4-regular augmentation for general graphs is NP-hard. We hope that the problem becomes solvable in P by limiting the given instances to 3-regular graphs. We began doing some research on this problem. Our preliminary results can be found in the Appendix A.

Conclusion. In conclusion, we can now say that — even though finding a planar and 3-regular augmentation is NP-hard in most cases — there are many cases in which we believe that finding planar and 3-regular augmentations are in P. We proved this for all 2-regular graphs and we assume that we can find planar and 3-regular augmentations for all graphs with maxmimum degree 2 in linear time. We also believe that there are at least some 3-regular graphs for which we can find a planar and 4-regular augmentation in P.

Bibliography

[Bat84] V. Batagelj. Inductive classes of cubic graphs. In *Finite and infinite sets*, volume 37 of *Colloquia mathematica Societatis János Bolyai*, pages 89–101. North-Holland, 1984.

[BGoM68] D.W. Barnette, B. Grünbaum, and Washington University Seattle Dept. of Mathematics. *On Steinitz's Theorem Concerning Convex 3-polytopes and on Some Properties of Planar Graphs.* Defense Technical Information Center, 1968.

[BKK+91] P. O. Bex, Goos Kant, Goos Kant, Hans L. Bodlaender, and Hans L. Bodlaender. Planar graph augmentation problems. In *In Proc. 2nd Workshop Algorithms Data Struct*, pages 286–298. Springer-Verlag, 1991.

[HRR12] Tanja Hartmann, Jonathan Rollin, and Ignaz Rutter. Cubic augmentation of planar graphs. In *ISAAC*, pages 402–412, 2012.

[HRR13] Tanja Hartmann, Jonathan Rollin, and Ignaz Rutter. Regular augmentation of planar graphs, 2013.

[NI02] Hiroshi Nagamochi and Toshihide Ibaraki. Graph connectivity and its augmentation: applications of ma orderings. *Discrete Appl. Math.*, 123(1-3):447–472, 2002.

[Ros64] M. Rosenfeld. Independent sets in regular graphs. *Israel Journal of Mathematics*, 2(4):262–272, 1964.

[SWK90] Wei-Kuan Shih, Sun Wu, and Yue-Sun Kuo. Unifying maximum cut and minimum cut of a planar graph. *IEEE Trans. Computers*, 39(5):694–697, 1990.

Appendix

A. Preliminary on Planar and 4-Regular Augmentations of Planar, 3-Regular and 3-Connected Graphs

We have began researching planar and 4-regular augmentations for planar, 3-regular and 3-connected graphs. We found out that for planar, 3-regular and 3-connected graphs the following necessary condition is always true:

Necessary Condition: Edge Permitting Faces Let G be a planar, 3-regular and 3-connected graph with n valencies and faces $f_1, f_2, \ldots, f_k$. A face f_i allows e_i edges running through its face. The sum is $\sum_{i=1}^{k} e_i \geq n/e$.

This is a necessary condition because a planar, 3-regular and 3-connected graph $G = (V, E)$ can only admit a planar and 4-regular augmentation if G possesses faces that allow to add $|V|/2$ edges (as we need to add $|V|/2$ edges in order to make a 3-regular graph 4-regular). A face with k adjacent vertices (we call it a *k-face*) allows $(k-2)/2$ edges. We say the face supplies $(k-2)/2$ edges. A face that is a triangle does not allow an edge running through the face. We can show that a planar, 3-regular and 3-connected graph always fulfills the necessary condition.

All planar, 3-regular and 3-connected graphs can be constructed by starting with K_4 and successively adding handles [BGoM68]. The K_4 does not admit a planar and 4-regular augmentation, but if we add a handle, the resulting graph admits such an augmentation and the faces allow enough edges (see Fig. A.1).

There are now only three possible ways to add new handles. We can either add a handle inside a face f supplying x edges such that two new faces f' and f'' are created that supply in total $z > x$ edges. This actually always occurs if neither f' nor f'' are triangles (see Fig. A.2).

Or we add a handle such that a new triangle face exists, there are two possible ways. Either the handle is added such that a new triangle face is created while another triangle becomes a 4-cycle. In this case, by adding the handle to the graph, the graph supplies one

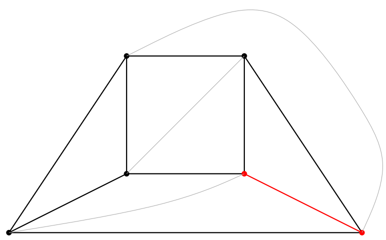


Figure A.1.: The resulting graph after adding a handle to K_4. The handle is shown in red. The planar and 3-regular augmentation is shown in grey

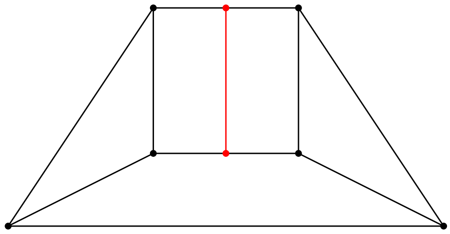
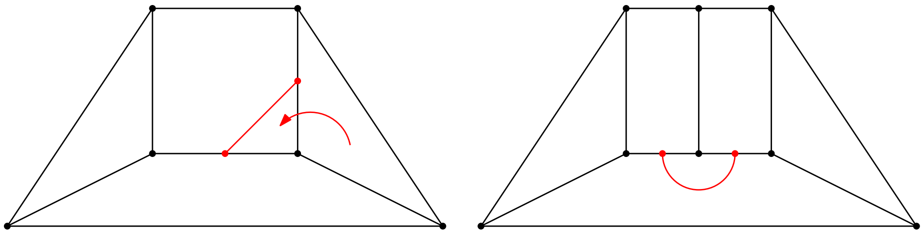


Figure A.2.: A handle is added such that the 4-face is split into two 4-faces supplying one edge more than before adding the handle.



(a) (b)

Figure A.3.: Planar, 3-regular and 3-connected graphs where the handle creates a triangle face.

edge more than before adding the handle. The faces permit enough edges add a 4-regular augmentation (see Fig. A.3 (a)).

Or we add a handle such that a new triangle face exists but no other triangle face becomes a 4-face as in Fig. A.3 (b), we can still show that the faces supply enough edges.

The smallest possible graph where we can add a handle such that a new triangle is created is the graph in Fig. A.4 as we need three 4-faces that are incident to a vertex. As the graph has to be 3-connected and 3-regular the remaining three vertices with valencies have to be connected to the same component. In case of the smallest graph, this component is only a vertex (in red). This graph provides faces that supply enough edges.

Also, because of 3-connectivity it is assured that every vertex is incident to at most one triangle face (except for the K_4). We hope that these necessary conditions are also sufficient conditions but we could not prove this yet.

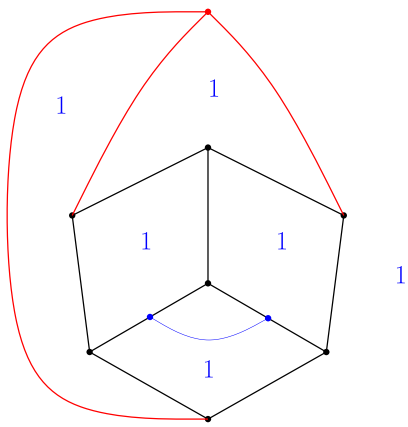
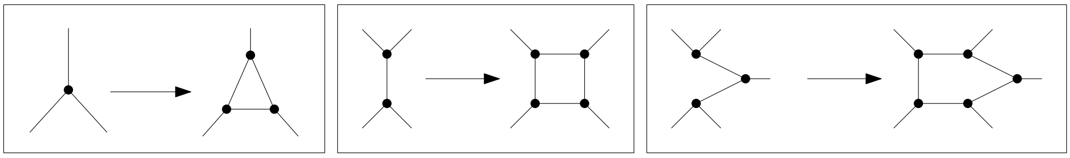


Figure A.4.: The smallest graph that allows a handle that creates a new triangle face without turning another triangle face into a 4-face.



(a) (b) (c)

Figure A.5.: Three expansion by Batagelj [Bat84] to create all planar, 3-regular and 3-connected graphs.

Besides adding handles, Batagelj shows another inductive way to create all planar, 3-regular and 3-connected graphs [Bat84] by expanding K_4 by three different components which are shown in Fig. A.5. Thus, we also looked for ways to find augmentations for all graphs by first finding an augmentation for the graph G that is shown in Fig. A.1. Afterwards we can expand the graph either by a handle as proved by Steinitz or expanding it after Batagelj resulting in a graph G' and modifying the augmentation of G such that we find an augmentation for G'. We hope with the help of the inductive classes of planar, 3-regular and 3-connected graphs that this is possible.

We assume that all planar, 3-regular and 3-connected graphs admit a 4-regular augmentation.