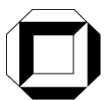Computing Large Matchings in Planar Graphs with
Fixed Minimum Degree

Autoren: Robert Franke, Ignaz Rutter,
Dorothea Wagner

Interner Bericht 2009-18

Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825

Fakultät für Informatik

Computing Large Matchings in Planar Graphs
with Fixed Minimum Degree

Robert Franke, Ignaz Rutter, Dorothea Wagner
`robert.franke@student.kit.edu`, `rutter@kit.edu`, `dorothea.wagner@kit.edu`

Abstract

In this paper we present algorithms that compute large matchings in planar graphs with fixed minimum degree. The algorithms give a guarantee on the size of the computed matching and run in linear time. Thus they are faster than the best known algorithm for computing maximum matchings in general graphs and in planar graphs, which run in $O(\sqrt{n}m)$ and $O(n^{1.188})$ time, respectively. For the class of planar graphs with minimum degree 3 the bounds we achieve are known to be best possible. Further, we discuss how minimum degree 5 can be used to obtain stronger bounds on the matching size.

1 Introduction

A *matching* is a set of independent (i.e., pairwise non-adjacent) edges in a graph. A *maximum* matching is a matching of maximum cardinality, and a *maximal* matching cannot be enlarged by adding edges.

The problem of finding maximum matchings in graphs has a long history, dating back to Petersen's theorem [19], which states that every biconnected 3-regular graph has a *perfect* matching, i.e., a matching that matches every vertex.

Finding maximum matchings, or large matchings in general, has many applications, see for example the book on matching theory of Lovász and Plummer [12]. To-date the asymptotically fastest (but rather complicated) algorithm for finding maximum matchings in general graphs runs in $O(\sqrt{n}m)$ time [13], where n and m are the numbers of vertices and edges of the given graph, respectively. Only recently faster algorithms for dense graphs, for planar graphs, for graphs of bounded genus, and for general H-minor free graphs have been suggested. They are all based on fast matrix multiplication (which, as a tool, is not very practical) and run in $O(n^\omega)$ time for dense graphs [15], $O(n^{\omega/2})$ time for planar graphs [16] and for graphs of bounded genus [27], and in $O(n^{3\omega/(\omega+3)}) \subset O(n^{1.326})$ time for H-minor free graphs [27], where $\omega \leq 2.376$ is the exponent in the running time of the best-known matrix-multiplication algorithm [5]. However, for practical purposes often slower, but less complicated algorithms are used: both LEDA [1] and the Boost Graph Library [23] provide maximum-matching algorithms that are based on repeatedly finding augmenting paths and have a running time of $O(nm\,\alpha(n,m))$ [24].

There has been a sequence of more and more general characterizations of graphs with perfect matchings [19, 8, 26]. This has also led to algorithms that test the existence of

or compute perfect matchings in $o(\sqrt{n}m)$ time in, e.g., bipartite k-regular graphs [22, 4], 3-regular biconnected graphs [2], and subgraphs of regular grids [25, 10, 11]. The last four algorithms all work in linear time for the corresponding subclasses of planar graphs. Moreover, for planar bipartite graphs a perfect matching can be computed in $O(n \log^3 n)$ time if it exists [14, 6]. There is also a fast algorithm for finding unique maximum matchings [7]. It takes $O(m \log^4 n)$ time in general and $O(n \log n)$ time in planar graphs.

There are combinatorial results that prove lower bounds on the size of maximum matchings in certain graph classes. Nishizeki and Baybars [17] show that planar graphs with minimum degrees 3, 4 and 5 have matchings of size at least $(n + 2)/3$, $(2n + 3)/5$ and $(5n + 6)/11$, respectively. Biedl et al. [3] show that maxdeg-3 graphs have a matching of size $(n - 1)/3$, 3-regular graphs have a matching of size $(4n - 1)/9$ and 3-connected planar graphs have a matching of size $(n + 4)/3$. However, these proofs are not constructive, in particular they do not indicate a way to find such a matching faster than by computation of a maximum matching. The only simple way to exploit these bounds algorithmically is to use the fact that a maximal matching (which can be computed quickly) has at least half the size of a maximum matching. The bounds obtained in this way are, however, rather weak, e.g., $(n + 2)/6$ for planar graphs with minimum degree 3 compared to the tight $(n + 2)/3$.

Recently, Rutter and Wolff [21] (a preliminary version appeared in [20]) gave fast algorithms that achieve the tight bounds of Biedl et al. Their algorithms compute matchings of size $(n - 1)/3$ in maxdeg-3 graphs in linear time, of size $(4n - 1)/9$ in 3-regular graphs in $O(n \log^4 n)$ time and of size $(n + 4)/3$ in 3-connected planar graphs in linear time. For graphs with bounded maximum degree k lower bounds for the size of maximal matchings where considered [9].

However, none of these results can be used to obtain matchings of guaranteed size in planar graphs with fixed minimum degree. In fact the question how fixed minimum degrees can be exploited algorithmically was posed as an open question in [21]. We answer this question and show that the tight bounds of Nishizeki and Baybars [17] for minimum degree 3 can be reached in linear time. We further analyze our algorithm in the context of minimum degree 5 and show that with some small modification it yields a matching of size $(2n + 1)/5$ in this case.

The bounds of Nishizeki and Baybars [17] for 1-connected planar graphs with minimum degree 3 were also obtained by Papadimitriou and Yannakakis [18]. They analyze the structure of maximum matchings and show that the structure is such that the free vertices can be balanced against the matching edges. We show that if we construct the matching accordingly, this balancing can be done locally: there is a pairing of free vertices with matching edges such that each free vertex is adjacent to its partner.

The paper is structured as follows. In Section 2 we propose a simple algorithm that already gives a non-trivial guarantee on the matching size, yet fails to reach the tight bound of $(n + 2)/3$ for planar minimum degree 3 graphs. We then analyze the algorithm and come up with additional structural conditions for the matching that allow us to improve its size. Section 3 then shows how these structural constraints can be employed to obtain a linear-time algorithm that finds matchings of size $(n+2)/3$ in planar graphs with minimum degree 3. We discuss how our approach can be generalized to obtain better bounds for planar graphs with minimum degree 5 in Section 4. We conclude and pose some open questions in Section 5.

2 Exploiting Minimum Degrees

In this section we describe a simple linear-time matching algorithm that already gives a non-trivial guarantee for planar mindeg 3 graphs. Our tight analysis then shows which aspects of the algorithm need to be improved in order to achieve the tight bounds of Nishizeki and Baybars [17].

We then show that certain additional structural requirements on the matching ensure that for minimum degree $\delta = 3$ we obtain the tight bound of Nishizeki and Baybars [17]. This analysis forms the basis of the algorithm presented in Section 3 where we show that a corresponding matching can be found quickly.

In order to present the algorithms we need some standard notation for graphs and matchings. Let $G = (V, E)$ be a graph and let M be a matching of G. We denote the *degree* of a vertex v by $d(v)$. A vertex in V is *free* (with respect to M) if it is not incident to an edge of M. An *augmenting path P* (with respect to M) is a path that alternates between edges in M and edges in $E \setminus M$ and starts and ends at different free vertices. In this case the symmetric difference of P and M is a matching of size $|M| + 1$. A matching is *k-free* if it does not admit an augmenting path of length up to k.

2.1 Algorithm Based on Short Augmenting Paths

We propose the following two-step algorithm MATCH3AUG: (1) Compute a maximal matching. (2) Iteratively find augmenting paths of length 3.

Lemma 1. *Let $G = (V, E)$ be a connected graph with m edges. MATCH3AUG computes a 3-free matching in $O(m)$ time.*

Proof. Step 1 is performed by choosing edges greedily. For Step 2 it is sufficient to consider the matching edges one by one and to check whether they are contained in an augmenting path of length 3. For an edge uv this can be done in $O(d(u) + d(v))$ time. The overall linearity follows from the fact that edges that are added to the matching during Step 2 need not be checked: Let xy be an edge that is added in Step 2 such that x was free after Step 1. If xy was contained in an augmenting path of length 3 then x would have a free neighbor x' contradicting the maximality of the matching after Step 1. □

In the following we analyze the size of 3-free matchings in planar graphs with minimum degree δ. To this end, we divide the free vertices into two disjoint sets that we bound independently.

Let $G = (V, E)$ be a planar graph with minimum degree δ and let M be a 3-free matching. Let $e \in M$ be an edge such that there is a free vertex $v \in V$ that is adjacent to both endpoints of e. We say that v *covers* e and that e is *covered*. An edge of the matching that is not covered by a vertex is *open*. Let M_C and M_O denote the set of covered and open edges of M, respectively. Moreover, let F_C denote the set of vertices that cover an edge and let F_O be the set of free vertices that do not cover any edge. Note that by definition M_C and M_O form a partition of M and F_C, F_O form a partition of the free vertices of V. Hence we have that $|M| = |M_C| + |M_O|$ and $n = 2 \cdot |M| + |F_C| + |F_O|$. We now bound the number of free vertices by independently bounding $|F_C|$ and $|F_O|$.

Lemma 2. *Let $G = (V, E)$ be a planar graph with minimum degree δ, let M be a 3-free matching and let M_C, M_O, F_C and F_O be defined as above. Then,*

$$|F_C| \quad \leq \quad |M_C| \tag{1}$$

$$|F_O| \quad \leq \quad 2 \cdot \frac{|M_O| - 2}{\delta - 2}. \tag{2}$$

Proof. First note that Equation (1) holds since the vertex covering an edge is unique as there would be an augmenting path of length 3 otherwise.

For the proof of Equation (2) consider the bipartite auxiliary graph $G' = (V', E')$ whose vertices are the vertices in F_O and the open edges of M. We connect a vertex $v \in F_O$ with an edge $m \in M_O$ if v is adjacent to an endpoint of m in G. The graph G' is planar as it can be obtained as a minor of G by removing all vertices that are either incident to an edge in M_C or cover an edge, contracting the remaining matching edges and removing edges that are not incident to a free vertex. Since no vertex of F_O is adjacent to an endpoint of a covered edge (there would be an augmenting path of length 3 otherwise), each vertex in F_O has degree at least δ in G'. Equation (2) now follows from $|E'| \leq 2 \cdot |V'| - 4 = 2 \cdot (|F_O| + |M_O|) - 4$ (bipartite, planar) and $|E'| \geq \delta|F_O|$ (minimum degree). $\square$

Theorem 1. *Let $G = (V, E)$ be a planar graph with n vertices, minimum degree $\delta \in \{3, 4\}$ and let M be a 3-free matching. Then the following holds:*

$$|M| \quad \geq \quad \frac{(\delta - 2) \cdot n + 4}{2 \cdot (\delta - 1)}. \tag{3}$$

Proof. Follows from Lemma 2 and $|V| = 2 \cdot |M| + |F_O| + |F_C|$. $\square$

Equation (3) does not hold for $\delta = 5$ as in this case the bound on $|F_C|$, which is independent of δ, is too weak. By Theorem 1 MATCH3AUG computes in linear time matchings of size at least $(n + 4)/4$ in planar graphs with minimum degree 3 and matchings of size $(n + 2)/3$ in planar graphs with minimum degree 4.

In order to obtain the bound $(n + 2)/3$ for $\delta = 3$ we would like to improve the bound of $|F_O|$ from Equation (2). However, Fig. 1 shows that our analysis is tight. Roughly speaking the problem is that the graph induced by the matching in this example is not connected. In the next section we give a precise definition of the desired property of the matching.

2.2 More Structure via Pure Tree-Like Matchings

Let $G = (V, E)$ be a planar graph with a fixed planar embedding, i.e., for every vertex v we have a cyclic ordering $\sigma(v)$ of its incident edges, and let M be a matching of G. Let G_M be the graph that is induced by the matched vertices of M. A matched vertex v is *cyclically pure* if its incident edges in G_M form an interval in $\sigma(v)$, further M is called *pure* if all matched vertices are cyclically pure. The matching M is called *tree-like* if G_M is a tree.

Lemma 3. *Let $G = (V, E)$ be a planar embedded graph and let M be a pure tree-like matching in G such that all free vertices have degree at least δ. Let F_M be the set of free vertices that have only matched neighbors. If F_M is not empty then there is a vertex $v \in F_M$ that has matched neighbors $x_1, \ldots, x_{\delta-2}$ such that each x_i has no other neighbor in F_M.*
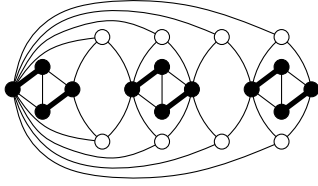
Figure 1: Planar graph with n vertices, min-deg 3 and a 3-free matching with only $(n+4)/4$ edges.
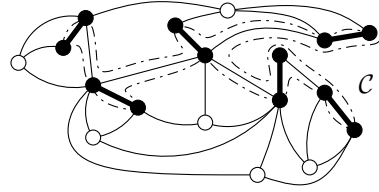

Figure 2: Construction of curve $\mathcal{C}$ that separates matched and free vertices of a pure tree-like matching.

Proof. In this proof we distinguish between *outer* vertices, i.e., matched vertices with free neighbors and *inner* vertices, i.e., matched vertices that are not outer. To prove the lemma we consider the subgraph G' of G that is induced by the edges that have one endpoint in F_M. We show that all outer vertices share a common face in the embedding inherited from G. Hence, by planarity, the vertices in F_M must have a parenthetical structure, where the most interior ones have the desired property.

Assume we have a planar drawing of G that realizes its given embedding. We construct a simple closed curve $\mathcal{C}$ that contains all outer vertices, encloses all inner ones and separates the matched vertices from the free vertices.

To construct $\mathcal{C}$ we pick an arbitrary outer vertex v as starting point and traverse the Euler tour of the tree G_M induced by the embedding and draw the curve along the edges of G_M. Beginning from v we traverse G_M starting along one of the edges bounding the interval of edges that connect v to a vertex in F_M. We position an imaginary pencil on the point where v lies in the plane and draw along the edges of G_M. Whenever we reach an outer vertex v' and would have to cross an edge that is incident to a free vertex to pass it, we instead draw right through v' and thus separate the free neighbors of v' from the matched ones. Since all vertices are cyclically pure the line visits every vertex at most once. Hence $\mathcal{C}$ is a simple cycle that contains every outer vertex. By construction $\mathcal{C}$ does not cross any edge and encloses all inner vertices but none of the free vertices (Fig. 2). Removing the interior of $\mathcal{C}$ (including edges) and free vertices that are not in F_M yields G' with all outer vertices sharing a common face.

There is at least one outer vertex that is incident with the outer face of G': Since all vertices in F_M are outside $\mathcal{C}$ there is at least one vertex x in F_M that is incident to the outer face of G'. Let e be an edge that is incident to x and bounds the outer face. The other endpoint of e is the desired vertex as it is matched and hence belongs to $\mathcal{C}$. Let $b_1, \ldots, b_k$ be the outer vertices as they occur along $\mathcal{C}$ in clockwise order, where b_1 is incident with the outer face of G_M.

Now consider the vertices of F_M (note that all their neighbors belong to $\mathcal{C}$). For each vertex v in F_M we set $b_\ell(v)$ and $b_r(v)$ to be the vertex b_i with the smallest, respectively largest, index i such that b_i is incident to v. We attach an opening parenthesis with label v to the edge $\{b_\ell(v), v\}$ for each v in F_M. Analogously we attach a closing parenthesis with label v to the edge $\{b_r(v), v\}$. We then traverse $b_1, \ldots, b_k$ and collect at each vertex all parentheses in clockwise order. This yields a sequence of opening and closing parentheses where matching opening and closing parentheses have the same label since a structure like $(_a \ (_b \ )_a \ )_b$ would contradict planarity.

Now pick a pair of parentheses that does not enclose any other parentheses. Let v be

the vertex that induces this pair. In addition to the two matched neighbors at which the parentheses were placed there have to be at least $\delta - 2$ further matched neighbors $x_1, \ldots, x_{\delta-2}$ since the degree of v is at least δ. Each of these vertices has only one free neighbor in F_M, namely v. $\square$

This result on pure tree-like matchings can be used to improve the bound on $|F_O|$ and hence the bound on 3-free matchings.

Lemma 4. *Let $G = (V, E)$ be a planar graph with minimum degree δ, let M be a pure tree-like 3-free matching and let M_O and F_O be defined as above. Then,*

$$|F_O| \quad \leq \quad \frac{|M_O| - 2}{\delta - 2}. \tag{4}$$

Proof. Let $\mathcal{C}$ be the curve from the proof of Lemma 3. Similar to the proof of Lemma 2 let G' be the graph obtained from G by contracting the edges in M_O, removing the vertices in F_C, the endpoints of edges in M_C and all edges in the interior of $\mathcal{C}$. This again yields a bipartite planar graph where one of the vertex sets corresponds to the open matching edges. Note that $\mathcal{C}$ is still a simple cycle since $\mathcal{C}$ contains at most one endpoint of each edge in M_O. However, since the interior of $\mathcal{C}$ is empty, the graph obtained from duplicating the vertices in F_O (together with their incident edges) is still planar and bipartite. Thus the bound is improved by a factor of 2, yielding the claim. $\square$

With the stronger bound of Equation 4 it follows that a pure tree-like 3-free matching in a planar graph with n vertices and minimum degree 3 has size at least $n/3$. For minimum degrees 4 and 5 the bound on $|F_C|$ is now weaker than the bound on $|F_O|$. Hence to obtain even stronger bounds we would need to improve the bound on the size of F_C.

Unfortunately, it is not easily possible to find a maximal matching that is both pure and tree-like in a given graph. Instead we show that we can construct such a matching by carefully removing free vertices when we cannot continue with enlarging the matching. The main part is to show that the number of removed vertices is bounded by the number of matching edges.

3 Algorithm

In this section we describe an algorithm that computes in linear time a matching of size at least $(n+2)/3$ in planar graphs with minimum degree 3. To show that our algorithm actually finds a matching of this size we use the following argument. In the course of the algorithm we perform a series of steps each of which either increases the size of the matching by 1 or deletes a free vertex. However, whenever a vertex is deleted, we make sure that there is an edge in the matching that "remembers" it in such a way that each matching edge "remembers" at most one vertex and no vertex is ever "forgotten". The algorithm finishes when there are no free vertices left. The bound then follows from the observation that there can be at most as many free vertices as matching edges.

The algorithm works as follows. We start by adding an arbitrary edge to the matching, which clearly is both pure and tree-like. We then enlarge the matching and make sure it remains pure and tree-like. To find an adequate spot to try to enlarge the matching we use Lemma 3: If there are only free vertices that also have free neighbors (i.e., $F_M = \emptyset$), we can easily find an edge that can be used to enlarge the matching, see Section 3.1. If F_M is not

empty (i.e., there are free vertices which have only matched neighbors) the lemma yields a free vertex v and a matched vertex x such that v is the only neighbor of x in F_M. In this case we try to enlarge the matching by two different strategies: a) If there is an augmenting path $vxyu$ of length 3, we will use this fact to swap xy for two new matching edges (Section 3.2). b) If x has free neighbors that have further free neighbors, we will use one of these and add an edge between two free vertices to the matching (Section 3.1).

In case neither of these strategies can be applied we remove v and show that there is a suitable matching edge that can remember it. The algorithm stops when no free vertices are left. In the following sections we describe these steps in detail and prove that they preserve a pure tree-like matching.

3.1 Enlargement by Adding a Suitable Edge

In this section we discuss how to enlarge a pure tree-like matching M by adding a suitable edge such that the outcome is still pure and tree-like. Consider a matched vertex x that has free neighbors and some of these have further free neighbors. Since the edges that connect x to free vertices form an interval in $\sigma(x)$, there exist a leftmost and a rightmost free neighbor of x (they coincide if x has only one free neighbor). To preserve cyclic purity we need that the leftmost or rightmost free neighbor u of x has a free neighbor u'. This situation occurs if x has at most one free neighbor that belongs to F_M and x is adjacent to a free vertex that is not in F_M, see Fig. 3a. The exact procedure is shown in the proof of the following lemma.

Lemma 5. *Let $G = (V, E)$ be a planar graph and let M be a pure tree-like matching in G such that each free vertex has degree at least δ. Further let x be a matched vertex such that the leftmost or rightmost free neighbor of x is adjacent to a free vertex. Then there is a graph $G' = (V, E')$ with $E' \subseteq E$ and a pure tree-like matching M' of G' such that $|M'| = |M| + 1$ and each free vertex has degree at least δ in G'.*

Proof. Without loss of generality, we can assume that the leftmost free neighbor u of x has a free neighbor. We now scan $\sigma(u)$ beginning with x until we find the first free neighbor u'. Let M' be $M \cup \{uu'\}$ and let G' be the graph that we obtain from G by removing all edges between u or u' and another matched vertex except for xu and uu'. We show that G' and M' satisfy the claim.

First, it is obvious that $|M'| = |M| + 1$ holds and each free vertex has the same degree as before since we only deleted edges that have both endpoints matched. It remains to show that M' is pure and tree-like. The vertex x is cyclically pure since u was the leftmost free neighbor of x and a possible edge xu' has been removed. Vertex u is cyclically pure, since it has just two matched neighbors x and u' and the edges ux and uu' are adjacent in $\sigma(u)$. Vertex u' is cyclically pure, because u is its only matched neighbor. The other matched vertices remain also cyclically pure as removing edges never violates cyclic purity. Thus M' is pure. Moreover, M' is tree-like since $G'_{M'}$ can be obtained by adding the branch xuu' to G_M (u and u' were free and thus not in G_M). $\qquad \square$

3.2 Exploiting Existence of an Augmenting Path of Length 3

In this section we describe how to make use of an augmenting path of length at most 3 in our context. Let $G = (V, E)$ be a planar embedded graph, let M be a pure tree-like matching
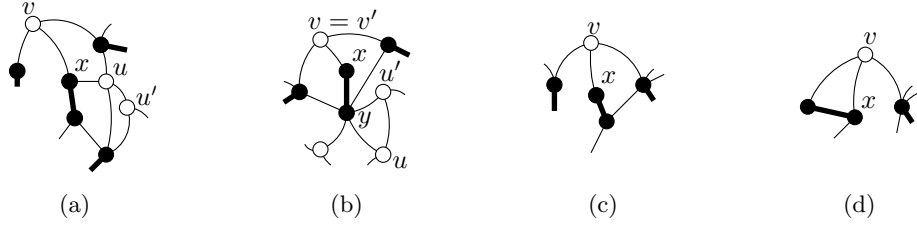
Figure 3: Illustration of the different cases that can occur in the algorithm for the candidate vertex x and its unique neighbor v in F_M.

in G such that all free vertices have degree at least 3 and let $vxyu$ be an augmenting path of length 3. We show that we can modify G and M such that M is enlarged by 1 and remains pure and tree-like. The problem is that just using the augmenting path to enlarge the matching may violate cyclic purity at the vertices u, v, x and y. Instead we show that there exists a suitable augmenting path of length 3 which leads (after removing some edges whose endpoints are both matched) to an enlarged pure tree-like matching. An example of this situation is shown in Fig. 3b.

Lemma 6. *Let $G = (V, E)$ be a planar graph and let M be a pure tree-like matching in G such that each free vertex has degree at least δ. Let $vxyu$ be an augmenting path of length 3. Then there is a graph $G' = (V, E')$ with $E' \subseteq E$ and a pure tree-like matching M' such that $|M'| = |M| + 1$ and each free vertex has degree at least δ in G'.*

Proof. Let x_ℓ and x_r be the leftmost and rightmost free neighbor of x, respectively, and define y_r, y_ℓ analogously. Choose $v' \in \{x_\ell, x_r\}$, $u' \in \{y_\ell, y_r\}$ such that v' and u' are distinct. This is always possible, otherwise $x_\ell = x_r = y_\ell = y_r$ and x and y both have only one free neighbor, which is actually shared by x and y, contradicting $v \neq u$.

We set $M' := (M \setminus \{xy\}) \cup \{v'x, yu'\}$ and let G' be the graph obtained from G by removing all edges that connect v' or u' to a matched vertex other than their matching partner. Clearly $|M'| = |M| + 1$ holds. We claim that M' is a pure tree-like matching in G' and every free vertex of G' has degree at least δ.

First note that by the choice of v' and u' the edges $v'x$ and yu' do not violate cyclic purity of x and y. The only edges that might violate cyclic purity at x and y are the edges $v'y$ and xv'. They are however removed when going from G to G'. Hence x and y are cyclically pure. By construction of G', v' and u' each have only one matched neighbor hence they are cyclically pure as well. All vertices different from v', x, y, u' may only have lost edges to free neighbors in G. Hence they all remain cyclically pure and M' is pure.

The graph $G'_{M'}$ is a tree since it can be obtained from the tree G_M by inserting the edges $v'x$ and $u'y$, which add the new leaves v' and u'. Thus M' is tree-like. Since we only remove edges that are not incident to free vertices the degrees of all free vertices are preserved. □

3.3 Linear-Time Algorithm

Lemma 5 and Lemma 6 yield together with Lemma 3 the simple algorithm MATCHMINDEG3, whose structure was outlined in the beginning of this section. A pseudo-code description of the algorithm is shown as Algorithm 1.

Algorithm 1 MATCHMINDEG3

1: Select an arbitrary edge e and set $M \leftarrow \{e\}$
2: **while** there are still free vertices **do**
3:   **if** $F_M \neq \emptyset$ **then**
4:     Select a matched vertex x and a free vertex v according to Lemma 3
5:     **if** there is an augmenting path $vxyu$ **then**
6:       Enlarge the matching according to Lemma 6
7:     **else if** x has a free neighbor outside of F_M **then**
8:       The leftmost or rightmost free neighbor of x suits to apply Lemma 5
9:     **else**
10:       Remove v (the matching edge that is incident to x remembers v)
11:   **else**
12:     Select a matched vertex that has free neighbors and apply Lemma 5

Theorem 2. *Let G be a planar embedded graph with n vertices and minimum degree 3. The algorithm* MATCHMINDEG3 *computes a matching of size at least $(n+1)/3$ in $O(n)$ time.*

Proof. We begin this proof by stating and justifying some loop invariants for the while-loop in MATCHMINDEG3.

(a) Each removed vertex is remembered by an adjacent matching edge

(b) Each matching edge remembers at most one vertex.

(c) The matching is pure and tree-like.

(d) Each free vertex has degree at least 3.

Invariants (a) and (b) are needed to prove the correctness of the algorithm while Invariants (c) and (d) ensure that the conditions of Lemmas 3, 5 and 6 are satisfied.

We now show that the algorithm preserves the invariants. When we delete a free vertex v, it is remembered by a matching edge xy (x and v are adjacent) that is not part of an augmenting path of length 3 and x has no other adjacent free vertices. Thus also y cannot be adjacent to a free vertex apart from v since there would be an augmenting path of length 3 otherwise. Hence xy will not have to remember another vertex and it is never removed from the matching (Fig. 3c, 3d). Thus Invariants (a) and (b) hold throughout the algorithm. Invariant (c) holds since we change the matching only by using Lemmas 5 and 6, which preserve the invariant. These lemmas together with the fact that we exclusively remove vertices that have only matched neighbors guarantee Invariant (d).

The size of the computed matching can now be seen as follows. Invariants (a), (b) and the observation that the last removed vertex has an additional remembering edge yields the bound $|F| \leq |M| - 1$ where F is the set of free vertices of G with respect to the output matching M. Using the equation $|F| = n - 2 \cdot |M|$ yields the bound $(n+1)/3 \leq |M|$.

Next, we discuss how to realize MATCHMINDEG3 in linear running time. In each iteration the number of free vertices is decreased by at least 1. Thus the algorithm stops after at most n iterations. Next, we show that each iteration of the while-loop runs in amortized $O(1)$ time.

For each vertex we store whether it is matched and if it has free neighbors. When a vertex v becomes matched it requires $O(d(v))$ time to propagate this information to its neighbors

such that they can update their number of free neighbors. The overall-time spent in this step is linear since a matched vertex remains matched (although its matching partner may change).

For matched vertices with free neighbors, we additionally store the first and the last edge leading to a free vertex. Note that given a matching edge we can hence easily check whether it is part of an augmenting path of length 3 since this involves only a constant number of vertices which can be found quickly using the first and last edge information. Note that the first and last edge can be updated in constant time when we remove an edge or match a free vertex. Moreover, for each matched vertex we store its F_M-*degree*, i.e., its number of neighbors in F_M. When the last free neighbor of a free vertex v gets matched or v is deleted, v notifies its neighbors, which then update their F_M-degree. Both cases occur at most once for each free vertex and thus this notification work needs linear time in total. By keeping a list of vertices with F_M-degree 1 we can find a candidate vertex x as in Lemma 3 in constant time.

The check whether Lemma 5 or Lemma 6 can be used to enlarge the matching can be done in $O(1)$ time: We only need to check a constant number of vertices for membership in F_M, which can be done by storing whether a vertex has free neighbors or not. The vertices we need to check can easily be addressed via the leftmost and rightmost free neighbor pointers.

The total time that is needed for all applications of the procedures provided by Lemma 6 and Lemma 5 is linear. This can be seen by considering occurrences of these cases. For applying Lemma 5 we first have to scan the neighborhood of a free vertex u for a free neighbor v, which requires $O(d(u))$ time. For the application of Lemma 6 the two vertices u and v that are newly matched can be identified in $O(1)$ time. In both cases in order to ensure the cyclical purity for the two newly matched vertices u and v we scan $\sigma(u)$ and $\sigma(v)$, which requires $O(d(u) + d(v))$ time. Since u and v are matched afterwards, they will not be processed in the same way again. Finally, removing a free vertex v can also be done in $O(d(v))$ time. ☐

Note that we can miss the tight bound of $(n + 2)/3$ by 1. We can, however, enlarge the matching by 1 in $O(n)$ time by computing an augmenting path [24].

4 A Better Bound for Minimum Degree 5

In this section we show how to improve the bound for $\delta = 5$ by ensuring that each removed vertex is remembered by more than one matching edge.

In case F_M is not empty, we previously considered a matched vertex x that had only one neighbor v in F_M. Now, instead, we consider more such matched vertices with the same neighbor in F_M at once. Either one of them can be used to enlarge the matching via Lemmas 5 and 6 or none of them and their matching partners are adjacent to another free vertex. In this case all their matching edges can be used to remember v. For $\delta = 5$ Lemma 3 yields a free vertex v that has three such neighbors and hence at least two matching edges can remember v.

Theorem 3. *Let $G = (V, E)$ be a planar graph with n vertices and minimum degree 5. A matching of size at least $(2n + 1)/5$ can be computed in $O(n)$ time.*

Proof. Let M be the matching computed by the modified algorithm. As shown above each free vertex F is remembered by at least two matching edges. Together with the observation

10

that the last free vertex is remembered by at least three edges we get $2 \cdot |F| + 1 \leq |M|$. The bound then follows from $n = 2 \cdot |M| + |F|$.

Next we show that the modified algorithm still runs in linear time. We need a way to maintain the set of candidate vertices with $\delta - 2$ matched neighbors according to Lemma 3. Instead of the F_M-degree of a matched vertex (i.e., the number of neighbors in F_M) we store a list of its neighbors in F_M. Whenever the length of this list changes to 1 or from 1 to another number we notify the (previously) last neighbor. This notification enables vertices in F_M to keep track of their neighbors with F_M-degree 1 by maintaining a list of them. Hence we can maintain a list of vertices in F_M that have at least $\delta - 2$ neighbors with F_M-degree 1. Thus we can pick such a candidate in $O(1)$ time and try to enlarge the matching using $\delta - 2$ of its F_M-degree 1 neighbors. The procedures given by Lemmas 5 and 6 are called at most $\delta - 2$ times per iteration. $\square$

5   Conclusion and Future Work

In this paper we have shown that it is possible to exploit minimum degrees in planar graphs algorithmically to obtain algorithms that compute matchings of guaranteed size quickly. Our algorithms run in linear time and yield matchings of size at least $(n + 2)/3$ and $(2n + 1)/5$ for planar graphs with minimum degrees 3 and 5, respectively.

While $(n+2)/3$ is tight for planar graphs with minimum degree 3, it is known that planar graphs with minimum degree 4 and 5 admit matchings of size $(2n + 3)/5$ and $(5n + 6)/11$, respectively. We leave open the question, whether these tight bounds can be achieved in linear time.

References

[1] Algorithmic Solutions.   The LEDA user manual version 5.2.   `www.algorithmic-solutions.info/leda_manual`, visited 07/04/2007.

[2] Therese Biedl, Prosenjit Bose, Eric Demaine, and Anna Lubiw. Efficient algorithms for Petersen's theorem. *J. Algorithms*, 38:110–134, 2001.

[3] Therese Biedl, Erik D. Demaine, Christian A. Duncan, Rudolf Fleischer, and Stephen G. Kobourov. Tight bounds on maximal and maximum matchings. *Discrete Math.*, 285(1–3):7–15, 2004.

[4] Richard Cole, Kirstin Ost, and Stefan Schirra. Edge-coloring bipartite multigraphs in $O(E \log D)$ time. *Combinatorica*, 21:5–12, 2001.

[5] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. In *Proc. 19th Annu. ACM Conf. Theory Comput. (STOC'87)*, pages 1–6, 1987.

[6] Jittat Fakcharoenphol and Satish Rao. Planar graphs, negative weight, shortest paths, and near linear time. *J. Comput. System Sci.*, 72:868–889, 2006.

[7] Harold N. Gabow, Haim Kaplan, and Robert E. Tarjan. Unique maximum matching algorithms. *J. Algorithms*, 40(2):159–183, 2001.

[8] Philip Hall. On representatives of subsets. *Jour. London Math. Soc.*, 10:26–30, 1935.

[9] Yijie Han. Matching for graphs of bounded degree. In *Frontiers in Algorithmics*, volume 5059 of *LNCS*, pages 171–173, 2008.

[10] Pierre Hansen and Mao Lin Zheng. A linear algorithm for perfect matching in hexagonal systems. *Discrete Math.*, 122(1-3):179–196, 1993.

[11] Claire Kenyon and Eric Rémila. Perfect matchings in the triangular lattice. *Discrete Math.*, 152(1–3):191–210, 1996.

[12] László Lovász and Michael D. Plummer. *Matching Theory.* North Holland, Amsterdam, 1986.

[13] Silvio Micali and Vijay V. Vazirani. An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matchings in general graphs. In *Proc. 21st Annu. IEEE Sympos. Found. Comput. Sci. (FOCS'80)*, pages 17–27, 1980.

[14] Gary L. Miller and Joseph Naor. Flow in planar graphs with multiple sources and sinks. *SIAM J. Comput.*, 24(5):1002–1017, 1995.

[15] Marcin Mucha and Piotr Sankowski. Maximum matchings via Gaussian elimination. In *Proc. 45th Annu. IEEE Sympos. Foundat. Comput. Sci. (FOCS'04)*, pages 248–255, 2004.

[16] Marcin Mucha and Piotr Sankowski. Maximum matchings in planar graphs via Gaussian elimination. *Algorithmica*, 45(1):3–20, 2006.

[17] Takao Nishizeki and Ilker Baybars. Lower bounds on the cardinality of the maximum matchings of planar graphs. *Discrete Math.*, 28(3):255–267, 1979.

[18] Christos H. Papadimitriou and Mihalis Yannakakis. Worst-case ratios for planar graphs and the method of induction on faces. In *Proc. 22nd Annu. IEEE Sympos. Foundat. Comput. Sci. (FOCS'81)*, pages 358–363, 1981.

[19] Julius Petersen. Die Theorie der regulären Graphs. *Acta Mathematica*, 15:193–220, 1891.

[20] Ignaz Rutter and Alexander Wolff. Computing large matchings fast. In *Proc. 19th Annu. ACM-SIAM Sympos. Discr. Algorithms (SODA'08)*, pages 183–192, 2008.

[21] Ignaz Rutter and Alexander Wolff. Computing large matchings fast. *Transactions on Algorithms*, to appear.

[22] Alexander Schrijver. Bipartite edge coloring in $O(\Delta m)$ time. *SIAM J. Comput.*, 28:841–846, 1999.

[23] Jeremiy Siek, Lie-Quan Lee, and Andrew Lumsdaine. The Boost Graph Library documentation. `www.boost.org/libs/graph`, visited 07/04/2007.

[24] Robert E. Tarjan. *Data structures and network algorithms.* SIAM, Philadelphia, 1983.

[25] William P. Thurston. Conway's tiling groups. *Amer. Math. Monthly*, 97(8):757–773, 1990.

[26] William T. Tutte. The factorization of linear graphs. *J. Lond. Math. Soc.*, 22:107–111, 1947.

[27] Raphael Yuster and Uri Zwick. Maximum matching in graphs with an exluded minor. In *Proc. 18th Annu. ACM-SIAM Sympos. Discr. Algorithms (SODA'07)*, pages 108–117, 2007.