COMPUTING THE GIRTH OF A PLANAR GRAPH IN $O(N \log N)$ TIME*

OREN WEIMANN† AND RAPHAEL YUSTER‡

Abstract. We give an $O(n \log n)$ algorithm for computing the girth (shortest cycle) of an undirected n-vertex planar graph. Our solution extends to any graph of bounded genus. This improves upon the best previously known algorithms for this problem.

Key words. Girth, shortest cycle, planar graph, graphs of bounded genus

AMS subject classifications. 05C38, 68R10

1. Introduction.

The *girth* of a graph is the length of its shortest cycle, or infinity if the graph does not contain any cycles. In addition to being a basic combinatorial characteristic of graphs, the girth has tight connections to many other graph properties. The connection between the girth of a graph and its chromatic number was studied by Erdős [13], Lovasz [19], Bollobás [4], and Cook [6]. Other important graph properties related to the girth include the minimum or average degree of the vertices, the diameter, the connectivity, the maximum genus, and the existence of certain type of minors (see Diestel's book [8] for a review of results).

The problem of computing the girth of a graph is among the most natural and easily stated algorithmic graph problems. Itai and Rodeh [17] were the first to suggest an efficient algorithm to compute the girth. They presented an $O(nm)$-time algorithm for a graph of n vertices and m edges, and an $O(n^2)$-time algorithm if an additive error of one is allowed. Monien [20] showed that finding the shortest cycle of even length is easier and can be done in $O(n^2\alpha(n))$ time, where $\alpha(n)$ is the inverse Ackermann function. Yuster and Zwick improved this to a pure $O(n^2)$ time algorithm [26]. Vazirani and Yannakakis [24] and Robertson *et al.* [23] studied the connection between such even-length cycles and Pfaffian orientations. Finding a cycle of a given size has also been extensively studied (see Alon *et al.* [1, 2] for references).

For the case of planar graphs, Eppstein [11] proved that the girth can be found in $O(n)$ time provided it is bounded by some constant. His result extended that of Itai and Rodeh [17] and of Papadimitriou and Yannakakis [21] who proved this for girth bounded by 3. For the general case, when the girth is not bounded by a constant, Djidjev [9] presented and algorithm that computes the girth in $O(n^{5/4} \log n)$ time. Djidjev's solution uses dynamic data structure for shortest paths [10], as well as a clever use of hammock decompositions [14]. Djidjev's algorithm is the fastest algorithm that solves this problem *directly*. However, there is another, *indirect* approach to solve the girth problem in planar graphs. It is a known fact that cuts in an embedded planar graph correspond to cycles in the *dual* plane graph. Furthermore, minimum cuts correspond to shortest cycles in the dual plane graph. Chalermsook *et al.* [5] gave an $O(n \log^2 n)$ time algorithm for the minimum-cut problem in planar graphs. This algorithm can be used to solve the girth problem in planar graphs with positive edge weights in the same time, by reducing it (in linear time) to the min-cut problem in planar graphs. We note that this reduction introduces (not necessarily constant) weights in the dual graph even if the original graph was unweighted.

---

*A preliminary version of this paper appeared in [25].

†Weizmann Institute of Science, Rehovot, 76100, Israel. (`oren.weimann@weizmann.ac.il`).

‡Department of Mathematics, University of Haifa, Haifa, Israel. (`raphy@math.haifa.ac.il`).

In this paper, we give an $O(n \log n)$-time algorithm for finding the girth of a planar graph. Apart from being faster than Djidjev's $O(n^{5/4} \log n)$ algorithm and from the $O(n \log^2 n)$ minimum-cut based algorithm, the structure of our algorithm is different – it is a simple divide-and-conquer, and we require no dynamic data structures. In addition, just like in Djidjev's case, our result extends from planar graphs to graphs of bounded genus. Unlike the minimum-cut based algorithm, or Djidjev's algorithms, we *do not* need to find an embedding of the graph in the plane (notice that any minimum-cut based algorithm must first embed the graph since it needs to construct the dual graph).

The rest of the paper is organized as follows. In Section 2 we recall some definitions and facts about planar graphs and bounded genus graphs. Sections 3 contains the description and proof of the algorithm for planar graphs, and Section 4 describes the generalization to bounded genus graphs. The final section contains some concluding remarks.

2. Preliminaries. A *planar embedding* of a graph assigns each vertex to a distinct point on the sphere, and assigns each edge to a simple curve between the points corresponding to its endpoints, with the property that the curves intersect only at their endpoints. A graph G is planar if it has a planar embedding. Consider the set of points on the sphere that are not assigned to any vertex or edge; each connected component of this set is a *face* of the embedding. A planar embedding on the sphere translates to a planar embedding in the plane where a chosen face becomes the outer face. If all the vertices of G lie on a single face, G is said to be *outerplanar* (or 1-outerplanar). G is *k-outerplanar* if the deletion of the vertices on the outer face results in a $(k-1)$-outerplanar graph.

The genus of a graph is the minimum number of handles that must be added to a sphere so that the graph can be embedded in the resulting surface with no crossing edges. A planar graph therefore has genus 0. Euler's formula states that a graph embedded on a surface of genus g with n vertices, m edges, and f faces, satisfies

$$n - m + f = 2 - 2g \; . \tag{2.1}$$

A separator is a set of vertices whose removal leaves no connected component of more than $2n/3$ vertices. If G is a planar graph, then it has a separator of $O(\sqrt{n})$ vertices [18], and if G has genus $g > 0$, then it has a separator of $O(\sqrt{gn})$ vertices [15]. The corresponding separators can be found in $O(n+g)$ time. Every k-outerplanar graph has a separator of size $O(k)$ [3, 22] that can be found in $O(n)$ time.

3. The Algorithm for Planar Graphs. In this section we prove the following main theorem of our paper.

THEOREM 3.1. *The girth of an undirected n-vertex planar graph can be computed in $O(n \log n)$ time.*

Given an embedded planar graph G, the size of each face of G is clearly an upper bound on G's girth. Notice however that the shortest cycle is not necessarily a face. Djidjev's algorithm [9], begins by computing, in $O(n)$ time, the size h of the minimal face of G. It then uses h to decide which of two procedures to apply in order to compute the girth. One procedure is used if h is below some specific threshold, and another if it is above. Our algorithm begins by computing, in $O(n)$ time, an *upper bound*, h, for the minimal face size of any embedding. We therefore avoid the need to compute an embedding explicitly. Unlike Djidjev's algorithm, our algorithm is a single divide-and-conquer procedure whose running-time is independent of h.

Our general idea is to cover G with $O(n/k)$ overlapping k-outerplanar graphs where $k = 2h$. The cover is constructed so that the smallest cycle in G is entirely contained within one of these k-outerplanar graph. This means that we can compute G's girth by independently computing the girth of each k-outerplanar graph. We use a simple algorithm on each k-outerplanar graph that exploits the fact that it has an $O(k)$ separator. We next describe this algorithm. In order to use it later we need the algorithm to work even if G's edges have positive edge lengths (and we seek the shortest, rather than smallest, cycle).

k-outerplanar Graphs with Nonnegative Edge-lengths. Given a k-outerplanar graph G with n vertices and nonnegative edge-lengths we describe an algorithm that computes G's shortest cycle in $O(kn \log n)$ time. The algorithm first constructs the $O(k)$-sized separator, and is then applied recursively on each of the connected components resulting from the removal of the separator. The recursive calls find G's shortest cycle in the case that it does not pass through any of the separator vertices. We are therefore left with finding G's shortest cycle in the case that it includes one or more of the separator vertices.

To do this, we first run a single-source shortest path algorithm from every separator vertex. Henzinger*et al.* [16] gave an $O(n)$-time algorithm for planar graphs with nonnegative edge-lengths that computes the distances from a given source v to all vertices of G. Therefore, in $O(kn)$ time, we can construct the shortest-path tree from every separator vertex. Suppose that the shortest cycle of G passes through some separator vertex v. The following lemma states an important connection between this cycle and the shortest-path tree from v to all vertices of G.

LEMMA 3.2. *Let G be a connected graph with positive edge-lengths. If a vertex v lies on a shortest cycle, and if T is a shortest paths tree from v then there is a shortest cycle that passes through v and has exactly one edge not in T.*

Proof. Suppose that the shortest cycle of G is of length s. Among all cycles of length s that pass through v, let C be the one with the least number of edges not in T. Assume for contradiction that this number is $k \geq 2$. A vertex $u \neq v$ on C partitions C into two parts C_1 and C_2 that are the two v-to-u simple paths in C. Since $k \geq 2$, there exists a vertex u so that both C_1 and C_2 contain an edge that is not in T. This is illustrated in Fig. 3.1

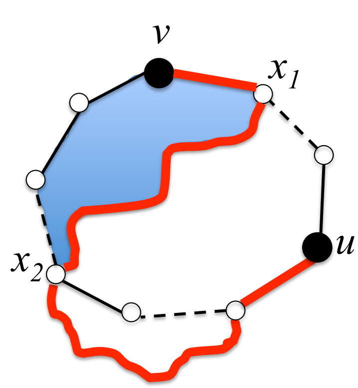

FIG. 3.1. *A cycle passing through a vertex v. The solid edges belong to the shortest-paths tree T and the dashed edges do not. The path P in bold (red) is the shortest path from v to u. The shaded area is a shorter cycle formed by a prefix of P and a part of the original cycle.*

Denote by P the path in T from the root v to the vertex u. Suppose that the only vertices that P and C_1 (resp. C_2) share are u and v. Then the cycle C' formed by P together with C_1 (resp. C_2) is of length at most s. This is because P is a shortest v-to-u path and thus not longer than C_2 (resp. C_1). However, C' has less than k edges that are not in T since all the edges of P are in T. This contradicts the fact that C is the shortest cycle with the least number of edges not in T.

Therefore, P must share some vertex $x_1 \notin \{u, v\}$ with C_1 and some vertex $x_2 \notin \{u, v\}$ with C_2. Without loss of generality we assume that x_1 appears before x_2 in P and that x_2 is the first vertex of P in $C_2 - \{u, v\}$. The prefix of P that ends in x_2, together with the part of C_2 between v and x_2, form a cycle C'. However, again, C' has less than k edges that are not in T, and this contradicts the fact that C is the shortest cycle with the least number of edges not in T. $\square$

The above lemma suggests the following $O(n)$-time procedure to find the shortest cycle in case it passes through v. Let T be the shortest-path tree rooted at v, and let $d_v(x)$ denote the length of the shortest path from v to x. For each edge (x, y) not in T whose length is $\ell(x, y)$ we look at $d_v(x) + d_v(y) + \ell(x, y)$ and take the minimum of this sum over all edges $(x, y) \notin T$.

Suppose the shortest cycle is of length s. Notice that if v is indeed part of a shortest cycle then by Lemma 3.2 we are guaranteed to find it using the above process. If on the other hand no shortest cycle passes through v then the value we get from this process is not smaller than s. This is because every value $d_v(x) + d_v(y) + \ell(x, y)$ that we consider corresponds to either an actual cycle or a cycle attached to a path (in the case where the shortest paths to x and to y share a common prefix). The $O(n)$ time complexity follows from the shortest paths algorithm of Henzinger *et al.* [16] and from the fact that the number of edges in G is $O(n)$ and each edge is checked in $O(1)$.

We have thus established that in $O(kn)$ time we can find the shortest cycle in the case that it passes through a separator vertex. If the removal of the separator results in $t \geq 2$ connected components, then the total time-complexity of all the recursive calls is therefore

$$T(n) = T(n_1) + T(n_2) + \cdots + T(n_t) + O(kn),$$

$$\text{where } \sum_{i=1}^{t} n_i \leq n \text{ and every } n_i \leq 2n/3 \ .$$

The solution to this recurrence is $T(n) = O(kn \log n)$ (for the standard analysis of such recurrences see, e.g. [7]).

This concludes our description of the k-outerplanar $O(kn \log n)$-time algorithm. Notice that this algorithm works even if the graph is directed. Indeed, suppose we want to compute the shortest directed cycle containing the separator vertex v. We start by deleting all the edges incoming to v. We then apply the Henzinger *et al.* algorithm (that works also for directed graphs) from source v. Let $d_v(x)$ denote the length of the shortest v-to-x path. We scan all edges (x, v) that we deleted before and take the minimum of $d_v(x) + \ell(x, v)$.

Since any planar graph G has a separator of size $\sqrt{|G|}$, our algorithm for directed planar graphs runs in time

$$T(G) = T(G_1) + T(G_2) + \cdots + T(G_t) + O(|G|^{3/2}),$$

$$\text{where } \sum_{i=1}^{t} |G_i| \leq |G| \text{ and every } |G_i| \leq 2|G|/3 \;.$$

This gives a total of $O(n^{3/2})$ time. We next show that in the undirected case this can be improved to $O(n \log n)$ by dividing the planar graph into many k-outerplanar graphs.

Covering the Graph by k-outerplanar Graphs. Before we can cover G by k-outerplanar graphs, we will need to modify G in order to make sure that each edge of G is incident with a vertex whose degree is *at least* 3. We note that this is opposite of Djidjev's algorithm [9] which makes sure that the maximum degree is 3. We may assume, of course, that G is 2-connected as otherwise we can run the algorithm on each 2-connected component separately. In particular, this implies that G has no vertices of degree 0 or degree 1. We may also assume that G is not a simple cycle as this case is trivial. We apply the following contraction to G repeatedly. We remove every vertex u of degree 2 whose two neighbors v_1, v_2 are not connected and add an edge (v_1, v_2) whose length is the sum of lengths of the edges (u, v_1) and (u, v_2). Once this contraction process ends we obtain a graph G' with the property that the girth of G is equal to the length of the shortest cycle in G'. Therefore, it suffices to compute the shortest cycle of G'. Notice that if h is the minimum face length of any embedding of G, then the number of edges on a shortest cycle of G' is also bounded by h. This is because the girth of G is bounded by h and we have only contracted edges to get from G to G'. The following lemma states two important properties of G'.

LEMMA 3.3. *In order to compute the girth of an n-vertex planar graph G, for which some embedding has minimum face length h, it suffices to compute the shortest cycle of the planar graph G', which has nonnegative edge-lengths and $O(n/h)$ vertices.*

Proof. Fix an embedding of G with minimal face length h. We will prove that the graph G' obtained by the above process has $O(n/h)$ vertices. We denote m as the number of edges in G, F denotes the set of all faces of G, $|x|$ denotes the size of a face $x \in F$ and f denotes the number of faces of G. Notice that the transformation from G to G' does not change the total number of faces. We will show first that $f = O(n/h)$. Since any edge of G belongs to two faces, then

$$2m = \sum_{x \in F} |x| \geq \sum_{x \in F} h = fh \;.$$

In any planar graph $m \leq 3n - 6$ so we get that $f \leq 2m/h \leq 6n/h = O(n/h)$.

Let m' and n' denote the number of edges and vertices of G'. We need to show that $n' = O(n/h)$, or, equivalently, that $m' = O(n/h)$. We denote T as the set of vertices of G with degree at least 3 and set $t = |T|$. As the set of vertices with degree 2 in G' is an independent set, we have that

$$\sum_{v \in T} deg(v) \geq m' \;.$$

On the other hand,

$$\sum_{v \in T} deg(v) + 2(n' - t) = 2m' \;.$$

By Euler's formula we know that

$$m' = n' + f - 2 \leq n' + 6n/h \;.$$

It follows that

$$\sum_{v \in T} (deg(v) - 2) = 2(m' - n') = 2f - 4 \leq 12n/h \ .$$

Since $deg(v) \geq 3$ for each $v \in T$ we have that

$$\frac{1}{3} \sum_{v \in T} deg(v) \leq 12n/h \ .$$

Consequently,

$$m' \leq 36n/h$$

as required. □

Lemma 3.3 actually provides a way to compute an upper bound h for the minimum face length of any embedding of G. We simply construct G' resulting in n' vertices, and set $h = \min\{n, \lfloor 36n/n' \rfloor\}$.

Now that we can work with a graph G' that has only $O(n/h)$ vertices we can finally describe how to cover G' by k-outerplanar graphs. Consider a breadth-first search of G' that starts in an arbitrary vertex r (and can be done in linear-time). Define G'_i as the graph induced by the vertices whose distance from r is between $ki/2$ and $k + ki/2$ for $k = 2h$ and $i = 0, 1, \ldots, \frac{2(n-k)}{k}$. In this way, every G'_i overlaps with at most two other graphs, G'_{i-1} and G'_{i+1}. This is depicted in Fig. 3.2. It is easy to verify that every G'_i is indeed a $(k+1)$-outerplanar graph. Furthermore, recall that the shortest cycle in G' has at most h edges. Therefore, it must be entirely contained within a single G'_i. This is because we chose k to be $2h$ and the overlap between two adjacent G'_i's to be $k/2$.

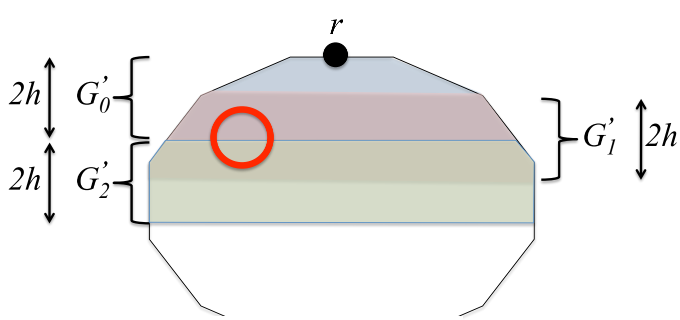

FIG. 3.2. *A decomposition of a graph G' into overlapping $2h$-outerplanar graphs according to a breadth-first search of G' that starts in an arbitrary vertex r. The shortest cycle is guaranteed to be completely contained within one of these $2h$-outerplanar graphs, in this case in G'_1.*

Finally, we run our k-outerplanar graph algorithm on every G'_i separately to find its shortest cycle. We then return the shortest cycle among these cycles. The time complexity is thus

$$\sum_i O(k|G'_i| \log |G'_i|) \leq O(2h \log n) \cdot \sum_i |G'_i| \ .$$

The $O(n \log n)$ total time complexity is achieved by noticing that every vertex in G' appears in at most three G'_i's therefore $\sum_i |G'_i| = O(|G'|)$, which is equal to $O(n/h)$ by Lemma 3.3. This completes the proof of Theorem 3.1.

4. Extension for Bounded Genus Graphs. In this section we show how to adjust the proof of Theorem 3.1 so that it extends to graphs with bounded genus. We therefore obtain the following theorem.

THEOREM 4.1. *For every fixed positive integer g, the girth of an undirected n-vertex graph whose genus is at most g can be computed in $O(n \log n)$ time.*
We outline the adjustments to the proof of Theorem 3.1 that are required in order to obtain Theorem 4.1. Regarding the shortest paths algorithm of Henzinger*et al.* [16] that we use, as pointed out in [16], separators of size $O(n^{1-\varepsilon})$ suffice for the application of their algorithm, provided that the separator can be found in linear time. Thus their algorithm remains $O(n)$ when applied to graphs with bounded genus.

In the proof of Theorem 3.1 the $(k+1)$-outerplanar graphs are just obtained by taking $k+1$ consecutive layers of a breadth-first search from a given vertex. We then use the fact that such graphs have $O(k)$-sized separators, and such separators are guaranteed to exist in subgraphs of these $(k+1)$-outerplanar graphs, as subgraphs of $(k+1)$-outerplanar graphs are also $(k+1)$-outerplanar. In other words, we simply use the fact that k-outerplanar graphs have *tree-width* $O(k)$. Now, suppose we perform breadth-first search in a genus g graph, and let G'_i be obtained by taking the $k+1$ consecutive layers s through $s+k$ of that search. We would like to claim that G'_i is analogous to a $(k+1)$-outerplanar graph in a "genus g" setting. Consider any embedding of the graph on a genus g surface. We can contract all vertices in layers above s to a single vertex z. The resulting graph is a minor of the original graph, thus the genus does not increase. Notice that now the diameter of G'_i becomes $O(k)$ and the genus remains at most g. A result of Eppstein [12] shows that graphs with bounded genus g have separators, as well as tree-width, of the same order as the diameter. It follows that G'_i has tree-width $O(k)$ as well, so the same analysis as in the case of k-outerplanar graphs holds in the bounded genus setting.

Another point of minor difference is in Euler's formula when applied in the proof of Lemma 3.3. Instead of using the fact that in planar graphs we have $m \le 3n-6$ we use the fact that in genus g graphs we have $m \le 3n-6+6g$. As g is bounded we still have $f = O(n/h)$ as in the planar case. Similarly, instead of using $m' = n' + f - 2$ which holds in the planar case we use $m' = n' + f - 2 + 2g$ and since g is bounded this still gives us that $m' = O(n/h)$ as in the planar case.

We have therefore shown that Theorem 3.1 can be adjusted to apply to the bounded genus setting, thereby proving Theorem 4.1.

5. Concluding Remarks and Open Problems. We have presented the fastest algorithm for computing the girth of an undirected planar graph and bounded genus graph. Our algorithm runs in $O(n \log n)$ time, improving the previous best algorithms. It would be interesting to extend this algorithm to undirected graphs with arbitrary positive *real* edge weights. It would also be interesting to find an $o(n^{3/2})$ algorithm for directed planar graphs.

REFERENCES

[1] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, 1995.
[2] N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17:354–364, 1997.
[3] H. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998.
[4] B. Bollobás. Chromatic number, girth and maximal degree. *Discrete Mathematics*, 24:311–314, 1978.

[5] P. Chalermsook, J. Fakcharoenphol, and D. Nanongkai. A deterministic near-linear time algorithm for finding minimum cuts in planar graphs. In *SODA*, pages 828–829, 2004.

[6] R.J. Cook. Chromatic number and girth. *Periodica Mathematica Hungarica*, 6:103–107, 1975.

[7] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. The MIT Press and McGraw-Hill Book Company, 2001.

[8] R. Diestel. *Graph theory*. Springer-Verlag, 2 edition, 2000.

[9] H. Djidjev. Computing the girth of a planar graph. In *Proceedings of the 27th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 821–831, 2000.

[10] H. Djidjev, G.E. Pantziou, and C.D. Zaroliagis. Improved algorithms for dynamic shortest paths. *Algorithmica*, 28(4):367–389, 2000.

[11] D. Eppstein. Subgraph isomorphism in planar graphs and related problems. *Journal of Graph Algorithms and Applications*, 3(3):1–27, 1999.

[12] D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3):275–291, 2000.

[13] P. Erdős. Graph theory and probability. *Canadian Journal of Math*, 11:34–38, 1959.

[14] G.N. Frederickson. Planar graph decomposition and all pairs shortest paths. *Journal of the ACM*, 38(1):162–204, 1991.

[15] J.R. Gilbert, J.P. Hutchinson, and R.E. Tarjan. A separator theorem for graphs of bounded genus. *Journal of Algorithms*, 5(3), 1984.

[16] M.R. Henzinger, P.N. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *Journal Comp. Sys. Sci.*, 55(1):3–23, 1997.

[17] A. Itai and M. Rodeh. Finding a minimum circuit in a graph. *SIAM J. Comput.*, 7(4):413–423, 1978.

[18] R.J Lipton and R.E. Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math*, 36:177–189, 1979.

[19] L. Lovasz. On chromatic number of finite set systems. *Acta Math. Acad. Sci. Hun.*, 19:59–67, 1968.

[20] B. Monien. The complexity of determining a shortest cycle of even length. *Computing*, 31:355–369, 1983.

[21] C.H Papadimitriou and M. Yannakakis. The clique problem for planar graphs. *Information Processing Letters*, 13:131–133, 1981.

[22] N. Robertson and P.D. Seymour. Graph minors. iii. planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984.

[23] N. Robertson, P.D. Seymour, and R. Thomas. Permanents, pfaffian orientations, and even directed circuits. *Ann. of Math.*, 150(3):929–975, 1999.

[24] V.V. Vazirani and M. Yannakakis. Pfaffian orientations, 0/1 permanents, and even cycles in directed graphs. In *Proceedings of the 15th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 667–681, 1988.

[25] O. Weimann and R. Yuster. Computing the girth of a planar graph in $o(n \log n)$ time. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 764–773, 2009.

[26] R. Yuster and U. Zwick. Finding even cycles even faster. *SIAM J. Discrete Math.*, 10(2):209–222, 1997.