PLANAR GRAPH BIPARTIZATION IN LINEAR TIME

SAMUEL FIORINI, NADIA HARDY, BRUCE REED, AND ADRIAN VETTA

Abstract. For each constant k, we present a linear time algorithm that, given a planar graph G, either finds a minimum odd cycle vertex transversal in G or guarantees that there is no transversal of size at most k.

1. Introduction

An *odd cycle transversal* (or *cover*) is a subset of the vertices of a graph G that hits all the odd cycles in G. Clearly the deletion of such a vertex set leaves a bipartite graph. Thus the problem of finding an odd cycle transversal of minimum cardinality is just the classical *graph bipartization problem.* Whilst this problem is NP-hard, it was recently shown [10] that an $O(n^2)$ time algorithm does exist when the size of an optimal solution is constant. This result is of particular interest given that in many practical examples, for example in computational biology [11], the transversals are typically small.

In this paper, we consider the restriction of the graph bipartization problem to planar graphs. As the vertex cover problem in planar graphs can be reduced to it, the restricted problem is still NP-hard. This and other related vertex and edge deletion problems in planar graphs have been extensively studied both structurally and algorithmically (see, for example, [8], [3] and [6]). Here we give a linear time algorithm for instances with constant sized optimal solutions. The graph properties of consequence in this problem are very different for planar graphs than for general graphs. By exploiting these properties, we develop an algorithm quite unlike that of [10].

We consider an embedding of the planar graph G. The *parity* of a face of G is defined as the parity of the edge set of its boundary, counting bridges twice. The crucial observation is that the parity of a cycle in G is equal mod 2 to the sum of the parities of the faces within it. In particular, it follows from the crucial observation that G is bipartite if and only if all its faces are even.

When a vertex v is deleted from G, all the faces incident to v are merged together in a new face F. The other faces are unchanged. We denote the new face by a capital letter to stress the fact that it determines a set of faces of G, namely, the faces of G included in it. Note that the parity of the new face F equals the sum mod 2 of the parities of the faces of G it contains. Let now W denote any set of

Key words and phrases. planar graph; odd cycle; linear time algorithm.

Samuel Fiorini is a Fellow of the Fonds de la Recherche Scientifique.

vertices in G. By deleting from G the vertices in W one after the other in some order, we see that each face of $G - W$ corresponds to a set of face of G (which is a singleton if the corresponding face is a face of G that survived in $G - W$). A face of $G - W$ is odd precisely if it contains an odd number of odd faces of G. Because a planar graph is bipartite if and only if all its faces are even, we obtain our

Key Fact: *A set W of vertices is an odd cycle cover of G precisely if every face of $G - W$ contains an even number of odd faces of G.* □

We remind readers that for a given embedding of G, the *face-vertex incidence graph* of G is the bipartite graph G^+ on the vertices and faces of G whose edges are the pairs fv, where f is a face of G and v is a vertex of G incident to f. In the next paragraph, we state a useful reformulation of the Key Fact in terms of T-joins in the face-vertex incidence graph. For the graph bipartization problem using edge deletions, Hadlock [4] considered a similar relationship between odd cycle (edge) transversals and T-joins in the dual graph. He used this to give a polynomial time algorithm for the maximum cut problem in planar graphs.

Consider any graph H and set of vertices T in H. A *T-join* in H is a set of edges J such that T equals the set of odd degree vertices in the subgraph of H determined by J. There exists a T-join in H if and only if each connected component of H contains an even number of vertices of T. In particular, if H has a T-join then $|T|$ is even. Now let T be the set of odd faces of the planar graph G. So T is an even set of vertices in the face-vertex incidence graph G^+. Letting $F(G)$ denote the set of faces of G, the correspondence between odd cycle transversals and T-joins is as follows.

Lemma 1. *A subset W of $V(G)$ is an odd cycle transversal of G if and only if the subgraph of G^+ induced on $W \cup F(G)$ contains a T-join, that is, every connected component of the subgraph has an even number of vertices of T.*

Proof. The lemma is equivalent to the Key Fact because deleting a vertex v from G corresponds to contracting all the edges incident to v in the face-vertex incidence graph G^+. □

The above lemma is useful because it enables us to visualize odd cycle transversals of G as forests in the face-vertex incidence graph G^+ such that each tree of the forest contains an even number of vertices of T. Indeed, consider an inclusionwise minimal odd cycle transversal W of G. By Lemma 1, i.e., by the Key Fact, there is a T-join J in G^+ covering each vertex of W and no vertex of $G - W$. Without loss of generality, we can assume that J is inclusionwise minimal. Then J is a forest and every leaf of J is in T. Note that some vertices of T can be internal nodes. From every vertex v of W, there are two internally disjoint paths in J from v to two elements of T.

So, letting $d_{\min}(v)$ be the minimum length of a path from v to an odd face in the face-vertex incidence graph, we see that the Key Fact implies:

Corollary 2. *No vertex v of G is in an inclusionwise minimal odd cycle cover of size less than $d_{\min}(v)$.* □

Thus, letting G' be the subgraph of G induced by $\{v|d_{\min}(v) > k\}$ we see that if G has an odd cycle cover of size at most k then G' must be bipartite. That is, $V(G) - V(G')$ is an odd cycle cover. So applying the Key Fact to the embedding of G' which appears as a sub-embedding of our embedding of G, we obtain:

Corollary 3. *If G has an odd cycle cover of size at most k then every face of G' contains an even number of odd faces of G.* □

We note further that the boundary, $bd(F)$, of every face F of G' is disjoint from the boundaries of the odd faces of G within it by the definition of G' (except for the trivial case $k = 0$). Thus we have:

Observation 4. *If G has an odd cycle cover of size at most k then there are at most k faces of G' which contain an odd face of G.* □

For some $r \leq k$, we let $\{F_1, ..., F_r\}$ be the set of faces of G' containing an odd face of G and let $G_i = G \cap (F_i \cup bd(F_i))$. Applying Corollary 2 again, it is easy to show:

Corollary 5. *If G has an odd cycle cover of order at most k then W is a minimum odd cycle cover of G precisely if $W_i = W \cap G_i$ is a minimum odd cycle cover of G_i for every i between 1 and r.*

Proof. Consider an odd cycle cover W of G of order at most k. By Corollary 2, W_i is disjoint from the boundary of F_i, and each face of $G - W$ which is not a face of G' is a face of $G_i - W_i$ for some i. Thus, applying the Key Fact to $G - W$ and $G_i - W_i$ for each i we see that W is an odd cycle cover of G if and only if W_i is an odd cycle cover of G_i for each i. Since W_i is disjoint from the boundary of F_i, the W_i's are disjoint and the result follows. □

It is easy to prove that the face-vertex incidence graph of each G_i has radius $O(k^2)$ and hence tree-width (defined below) which is $O(k^2)$. We show in Section 3 that we can find minimum odd cycle transversals in linear time in graphs with bounded tree-width. So if we could find all the G_i in linear time then we could compute a minimum odd cycle cover for each G_i in linear time and by taking their union, find a minimum odd cycle cover of G (or determine that G has no odd cycle cover of order at most k). This is close to what we do. There is one slight technical hitch, we actually need to consider a subgraph G'' of G'. We give details in the next section.

We close this introductory section with some more remarks on related work concerning odd cycle packing and covering in planar graphs. Reed [9] showed that the following Erdös-Pósa property holds in planar graphs: for any integer k, there exists an $f(k)$ such that G either has an odd cycle transversal of size at most $f(k)$ or a packing of vertex disjoint odd cycles of size at least $k+1$. For the edge version of this problem, Král and Voss [7] recently proved that the Erdös-Pósa property holds for $f(k) = 2k$, which is a tight result. In contrast, it is easy to show that in general graphs the Erdös-Pósa property does not hold.

2. The Algorithm

Our algorithm works as follows. First obtain an embedding of G in linear time [5], and construct the face-vertex incidence graph G^+. Then find a collection $\mathcal{F} = \{f_1, \ldots, f_s\}$ of boundary-disjoint odd faces of G which either has $k+1$ faces or is inclusion-wise maximal. This part of the algorithm can be implemented to run in $O(kn)$ time which is linear as k is fixed.

If $s = k+1$ then return the information that G has no odd cycle transversal of size at most k and stop. Otherwise, let B_i denote the set of faces and vertices of G whose distance to f_i in G^+ is at most $k+1$. Determine the sets B_i for all $i = 1, \ldots, s$ via a breadth first search in G^+. Let G'' be the subgraph of G obtained by deleting all the vertices of G in each B_i. We note that G'' is a subgraph of G' because every odd face is incident to some f_i.

Determine the set $F_1, ..., F_r$ of faces of the embedding of G'' which contain an odd face of G. Note that $r \le s \le k$ as each F_i contains some f_j. We let D_i be the subgraph of G contained in the union of F_i and its boundary. We refer to these graphs as *discs*. Now find a minimum odd cycle transversal W_i in each disc D_i. Since, as we show below, each disc has bounded tree-width, this can be achieved in linear time using the techniques described in Section 3. Let W be the union of $W_1, \ldots, W_r$. If W has size at most k, then W is a minimum odd cycle transversal of G; output W. Otherwise, return the information that G has no odd cycle transversal of size at most k. This concludes the description of the algorithm.

Proposition 6. *The algorithm finds a minimum cardinality odd cycle transversal if G has an odd cycle transversal of size at most k or otherwise detects that no such transversal exists.*

Proof. The proof of this proposition mimics exactly the proof of Corollary 5 with G' replaced by G'' and G_i replaced by D_i. □

In Section 3, we will describe how to find minimum odd cycle transversals in graphs of bounded tree-width in linear time. Since all of the steps described in this section can be carried out in linear time, Proposition 6 tells us that we will obtain a linear time algorithm for general planar graphs if we can show that each disc has bounded tree-width. This, though, follows simply from the following result.

Lemma 7. *([1], for a more general result see [12, 13]) If a planar graph contains no $h \times h$ grid minor, then its tree-width is at most $8h$.* □

Because the radius of the face-vertex incidence graph of any planar graph containing a $h \times h$ grid minor is at least h, the preceding lemma has the following immediate corollary:

Corollary 8. *Let G be a planar graph. If the radius of the face-vertex incidence graph of G is less than h, then the tree-width of G is at most $8h$.* □

Lemma 9. *The tree-width of each disc is $O(k^2)$.*

Proof. By Corollary 8, it suffices to show that the radius of each disc is $O(k^2)$. Consider a disc D_i and let f_j denote any member of $\mathcal{F}$ which is a face of D_i. We claim that any face of vertex of D_i is within a distance of at most $2k^2 + 3k - 2$ from f_j in the face-vertex incidence graph of D_i. Let I be the set of indices ℓ such that f_ℓ is a face of D_i. Let H denote the graph on I in which two indices ℓ and ℓ' are adjacent if there is a path of length at most 2 from an element of B_ℓ to an element of $B_{\ell'}$ in G^+. Then the distance in D_i^+ between any f_ℓ with $\ell \in I$ and f_j is at most the distance in H between ℓ and j times $2(k+1)+2 = 2k+4$. Moreover, for every face or vertex of D_i there is an index $\ell \in I$ such that the distance in D_i^+ between the considered face or vertex and f_ℓ is at most $k+2$. Because H has at most k vertices and is connected, the distance in D_i^+ between any face or vertex of D_i and f_j is at most

$$(2k+4)(k-1) + k + 2 = 2k^2 + 3k - 2.$$

So the claim holds and the radius of D_i^+ is at most $2k^2 + 3k - 2 = O(k^2)$. □

3. Odd Cycle Transversals in Graphs of Bounded Tree-width

As we have seen, it suffices to find a linear time algorithm for graphs with bounded tree-width. This can be done using standard techniques; we present such an algorithm below. Our main result then follows.

We begin with the required technical definitions. A *tree-decomposition* of G is a pair $(T, \mathcal{V})$, where T is a tree and $\mathcal{V} = (V_t \subseteq V(G) : t \in V(T))$ is a family of subsets of $V(G)$ with the following properties:

(1) $\bigcup(V_t : t \in V(T)) = V(G)$.
(2) For each edge $e \in E(G)$ there is a $t \in V(T)$ such that both endpoints of e are in V_t.
(3) For t_0, t_1 and t_2 in $\in V(T)$, if t_0 is on the path of T between t_1 and t_2, then $V_{t_1} \cap V_{t_2} \subseteq V_{t_0}$.

The *width* of the tree-decomposition $(T, \mathcal{V})$ is defined as $\max_{t \in V(T)}(|V_t| - 1)$. The *tree-width* of a graph G is the minimum w such that G has a tree-decomposition of width w. It is well known that there are minimum tree decompositions of G that use at most n nodes. Moreover, we can easily convert a tree decomposition $(T, \mathcal{V})$ to another $(T', \mathcal{V}')$ of the same width, such that T' is a binary tree with at most twice as many nodes as T. Let G be a graph with bounded tree-width $\omega - 1$

and let $(T, \mathcal{V})$ be a binary minimum tree-decomposition of G. We denote by t the nodes of T and by V_t the subset of $V(G)$ assigned to t. We have that $|V_t| \leq \omega$ for all $t \in T$. Pick an arbitrary root node $t^* \in T$. Then, given a node $t \in T$ we let S_t be the subtree of T rooted at t. From (2) we may assign to each edge $e = (u, v)$ of G a specific node $t(e) \in T$ for which $u, v \in V_t$. Thus, for each $t \in T$ there is an associated edge set $E_t \subseteq E(G)$. Hence, we may define the graphs $G(t) = (V_t, E_t)$ and $G(S_t) = (\bigcup_{t' \in S_t} V_{t'}, \bigcup_{t' \in S_t} E_{t'})$.

We associate with each node $t \in T$ a set $\mathcal{A}_t$ of all the ordered triplets $\Pi_t = (L_t, R_t, W_t)$ where L_t, R_t and W_t form a vertex partition of V_t. Clearly $|\mathcal{A}_t|$ is at most 3^ω. Our algorithm will work up from the leaves maintaining the property that for each partition Π_t we (implicitly) store a minimum odd cycle transversal $\hat{W}_t$ in $G(S_t)$ that is *accordant* with the partition. That is, $W_t \subseteq \hat{W}_t$ and L_t and R_t are on opposites sides of the bipartition in $G(S_t) - \hat{W}_t$. If such a transversal exists then we will set $f(\Pi_t) = |\hat{W}_t|$; otherwise if there is no such accordant transversal then we set $f(\Pi_t) = \infty$. Hence, for a leaf $t \in T$ we have $f(\Pi_t) = |W_t|$ if L_t and R_t both induce stable sets in E_t. Otherwise $f(\Pi_t) = \infty$. Now take a non-leaf node $t \in T$ with children r and s. If L_t or R_t induce an edge in E_t then we set $f(\Pi_t) = \infty$. So suppose not. We say that a partition $\Pi_r = (L_r, R_r, W_r)$ in $\mathcal{A}_r$ is *consistent* with a partition $\Pi_t = (L_t, R_t, W_t)$ in $\mathcal{A}_t$ if $W_t \cap V(S_r) \subseteq W_r$, $L_t \cap V(S_r) \subseteq L_r$ and $R_t \cap V(S_r) \subseteq R_r$. We use the notation $\Pi_r \sim \Pi_t$ to denote consistency. Note, by property (3), that if Π_r and Π_s are both consistent with Π_t then they are consistent with each other. Then set

$$f(\Pi_t) = \min_{\Pi_r \sim \Pi_t, \Pi_s \sim \Pi_t} f(\Pi_r) + f(\Pi_s) + |W_t - (W_r \cup W_s)| - |W_r \cap W_s|$$

Note that it may still be the case that $f(\Pi_t) = \infty$. We repeat this process up the tree. Observe that, by storing pointers from a partition Π_t to the partitions Π'_r and Π'_s in its children that produced the minimum value $f(\Pi_t)$, we may implicitly store the set $\hat{W}_t$. We then obtain the following result.

Lemma 10. *For each Π_t, either $f(\Pi_t)$ is the size of the minimum odd cycle transversal in $G(S_t)$ accordant with the partition Π_t, or $f(\Pi_t) = \infty$ and no such a transversal exists.*

Proof. This is clearly true if t is a leaf. So let $t \in T$ be a non-leaf with children r and s. Take Π_t and assume first that $f(\Pi_t)$ is finite. Next take consistent partitions Π_r and Π_s with optimal transversals $\hat{W}_r$ and $\hat{W}_s$, respectively. Then, since $\hat{W}_r$ and $\hat{W}_s$ are accordant with Π_r and Π_s, by property (3) we have that $W_t - (\hat{W}_r \cup \hat{W}_s) = W_t - (W_r \cup W_s)$. Thus, in obtaining $\hat{W}_t$ we only need to add the vertices in $W_t - (W_r \cup W_s)$. Moreover any vertex in $W_r \cap W_s$ is double counted by $f(\Pi_r) + f(\Pi_s)$. Thus $f(\Pi_t)$ is in fact the size of a transversal in $G(t)$ accordant with Π_t. Therefore, since we are examining all consistent pairs of partitions for the children, it is clear that $f(\Pi_t)$ is the size of a minimum odd cycle transversal

$\hat{W}_t$ in $G(S_t)$ accordant with the partition Π_t. Now suppose $f(\Pi_t) = \infty$ and that there is a transversal W for $G(S_t)$ accordant with Π_t. Then, for all pairs of partitions Π_r and Π_s that are consistent with Π_t, at least one of $f(\Pi_r)$ or $f(\Pi_s)$ is infinite. We obtain a contradiction as the restrictions of W to $G(S_r)$ and $G(S_s)$ give odd cycle transversals for these subgraphs that are accordant with Π_r and Π_s, respectively. □ □

It immediately follows that the minimum transversal can be found by considering the partition Π_{t^*} with the minimum f value. We may obtain a binary tree-decomposition in linear time [2]. For each node in the tree we have $O(3^\omega)$ partitions. It takes $O(|E_t|)$ time to check whether L_t or R_t induce stable sets in $G(t)$. There are then $O(9^\omega)$ possible pairs of partitions for the children. Thus it takes $O(\omega 9^\omega)$ time to check for consistencies and to calculate $f(\Pi_t)$. In total, therefore the algorithm runs in time $O(\omega 3^{3\omega} n)$. Thus we have proven Theorem 11 and Corollary 12.

Theorem 11. *Let G be a graph with bounded tree-width. Then there is an linear time algorithm to find a minimum odd cycle transversal in G.* □

Corollary 12. *In a planar graph, for any constant k, there is an $O(n)$ time algorithm to find a minimum odd cycle transversal of cardinality at most k or determine that no such transversal exists.* □

References

[1] N. Alon, P. Seymour and R. Thomas, "Planar separators", *SIAM J. Discrete Math.*, **7(2)**, pp184-193, 1994.

[2] H. Bodlaender, "A linear time algorithm for finding tree-decompositions of small treewidth", *SIAM Journal on Computing*, **25**, pp1305-1317, 1996.

[3] M. Goemans and D. Williamson, "Primal-dual approximation algorithms for feedback problems in planar graphs", *Combinatorica*, **18**, pp37-59, 1998.

[4] F. Hadlock, "Finding a maximum cut of a planar graph in polynomial time", *SIAM J. Comput.*, **4(3)**, pp221-225, 1975.

[5] J. Hopcroft and R. Tarjan, "Efficient planarity testing", *J. Assoc. Comput. Mach*, **21**, pp549-568, 1974.

[6] J. Kleinberg and A. Kumar, "Wavelength conversion in optical networks". *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms (Baltimore, MD, 1999)*, pp566-575, 1999.

[7] D. Král and H. Voss, "Edge-Disjoint Odd Cycles in Planar Graphs", *Journal of Combinatorial Theory Series B*, **90(1)**, pp107-120, 2004.

[8] C. Lucchesi and D. Younger, "A minimax theorem for directed graphs", *J. London Math. Soc.*, **17**, pp369-374, 1978.

[9] B. Reed, "Mangoes and blueberries", *Combinatorica*, **19**, pp267-296, 1999.

[10] B. Reed, K. Smith and A. Vetta, "Finding odd cycle transversals", **32**, *Operations Research Letters*, pp299-301, 2004.

[11] R. Rizzi, V. Bafna, S. Istrail and G. Lancia, "Practical algorithms and fixed parameter tractability for the single individual SNP haplotyping problem", *Algorithms in Bioinformatics*, LNCS **2452**, Springer, pp29-43, 2002.

[12] N. Robertson and P. Seymour, "Graph Minors V. Excluding a planar graph", *J. Combin. Theory Ser. B*, **36**, pp92-114, 1986.
[13] N. Robertson, P. Seymour and R. Thomas, "Quickly excluding a planar graph", *J. Combin. Theory Ser. B*, **62**, pp323-348, 1994.