CS762: Graph-Theoretic Algorithms
Lecture 27: k-outerplanar graphs
March 20, 2002

Scribe: Therese Biedl

Abstract

We introduced k-outerplanar graphs in the last lecture. In this lecture, we show that they are related to the graphs of bounded treewidth (more precisely, a k-outerplanar graph has treewidth at most $4k+2$).

1 Introduction

We defined k-outerplanar graphs in the previous lecture. Loosely speaking, they are defined as follows. Let G be a planar graph with a fixed planar embedding. Repeatedly "peel" a layer off G, i.e., repeatedly remove all vertices that are currently on the outer-face. G is called *k-outerplanar* if, for some planar embedding of G, we are left with nothing after peeling at most k times.

In this lecture, we will show that k-outerplanar graphs have treewidth at most $4k+2$. (In fact, one can show a tighter bound – they have treewidth at most $3k-1$ – but we will not do this here.) In consequence, every problem that can be solved efficiently for partial k-trees can also be solved efficiently for k-outerplanar graphs. Such problems include Independent Set, Hamiltonian Cycle, and others.

The proof in this lecture follows loosely the exposition in [Bod88].

2 Bounded treewidth

The objective of this section (and in fact, almost all of this lecture) is to show the following theorem:

Theorem 1 *Every k-outerplanar graph G has treewidth at most $4k+2$.*

The proof of this theorem breaks down into three steps.

- Reduce the problem to graphs of maximum degree 3. Thus, show that there exists a k-outerplanar graph G' that has maximum degree 3 such that treewidth(G) $\leq$treewidth(G').
- Show that every k-outerplanar graph with maximum degree 3 has a spanning tree with load at most $2k$. (We will define load formally later.)
- Show that if a graph with maximum degree 3 has a spanning tree with load ℓ, then it has treewidth at most $2\ell+2$.

2.1 Achieving maximum degree 3

So the first step is to modify G into a graph G' that has maximum degree 3, is k-outerplanar and treewidth$(G) \leq$treewidth(G').

We will first show how to modify one vertex of high degree. Assume that G is a k-outerplanar graph, and v is a vertex in G that has $\deg(v) \geq 4$. Then we replace v by a chain $C(v)$ of $\deg(v)-2$ vertices. Each endpoint of the chain is incident to two edges that used to be incident to v, and all other vertices of the chain are incident to one edge that used to be incident to v. See Figure 1 for an illustration.

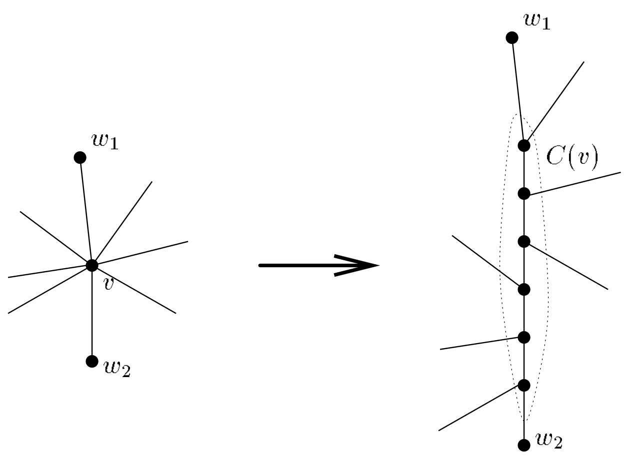

Figure 1: Replacing v by a chain $C(v)$ of vertices.

Incidentally, this replacement of a vertex of high degree by a chain of vertices of degree 3 is a common technique, also used in Graph Drawing, or for showing that problems remain NP-complete even in graphs with small maximum degree. In order for it to do the job for k-outerplanar graphs, we need to be careful in which edges we assign where along the chain. We need the following two rules:

- Since the graph was planar, there was a fixed ordering of the edges around vertex v. Maintain this ordering while splitting vertex v into a chain of vertices.
- Choose edges incident to the ends of the chain in a special way. More precisely, consider the time when we had peeled away just enough layers from the graph such that v is on the outer-face. Then let w_1 and w_2 be two neighbours of v on the outer-face of the current graph. Assign the edges to the chain in such a way that the edges (w_1, v) and (w_2, v) are incident to the two ends of the chain.

Let G' be the graph that results from G by replacing all vertices of degree at least four in such a fashion. Clearly G' has maximum degree 3. It is also not hard to see that G' is again k-outerplanar, because the layer that used to contain v now contains $C(v)$ (by our choice of w_1 and w_2 at the ends of the chain), and hence all vertices of $C(v)$ are deleted at the same time that we would have deleted v.

So all that remains to show is that treewidth$(G) \leq$treewidth(G'). To show this, observe that G can be obtained from G' by contracting all edges between vertices in $C(v)$ (for every vertex v). Recall the operation of *contracting an edge*. If G is a graph with an edge (x, y), then by contracting the edge we mean that we delete edge (x, y) and vertices x and y, and introduce a new vertex z.

All edges that were previously incident to x or y are now incident to vertex z. See also Figure 2. If the graph is planar, then we maintain the planar embedding; more precisely, the clockwise order around z is first all edges that were incident to x (in the order that they were in G), and then all edges that were incident to y (also in the order in which they were in G).

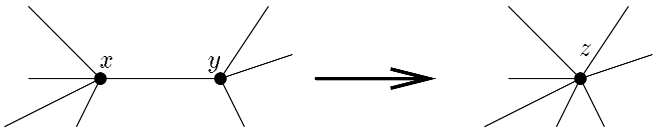

Figure 2: The operation of contracting an edge.

So to show that treewidth(G) $\leq$treewidth(G'), the following lemma suffices:

Lemma 1 *Let G be a graph that is obtained from graph H by contracting edge (x,y) of H into a vertex z in G. Then treewidth(G) $\leq$treewidth(H).*

Proof: Assume we have a tree decomposition of H. Replace every occurrence of either x or y with z. We claim that this is a tree decomposition of G, and must verify the three properties of a tree decomposition:

- Every vertex of G appears in at least one label. This clearly holds, since every vertex of H appeared, and vertex z appears at least once since x and y did.
- For every edge in G, there exists a label that contains both endpoints. Again, this clearly holds because it held for the tree decomposition of H. The only edges where there was a change are those incident to x and y. For those edges, we have a new endpoint z, but we also replaced x or y by z, so the claim still holds.
- For every vertex in G, the set of labels containing the vertex is connected. This clearly holds for all vertices in G that are also in H, so we only have to show it for z. Now, the labels containing x form a subtree, and so do the labels containing y. Moreover, there was an edge (x,y) in H, and hence a label that contains both x and y. Therefore, the union of these two subtrees is also connected, and that is exactly the subtree for z.

Since we did not add to any label, the treewidth is the same for both tree decompositions, and the treewidth of G can only be better than that, which yields the result. $\square$

2.2 Spanning trees of small load

Assume that G is a connected graph, and T is a spanning tree of G. Let e be an edge in the tree. Then the *load* of the tree-edge e is the number of non-tree edges (x,y) in G such that the unique path between x and y in T uses edge e. See Figure 3 for an example. (In some references, the load is also called the *remember number.*) The *load* of tree T is the maximum load of an edge.

For the induction hypothesis to work out properly in the proof to come, we need to allow the graph to become disconnected. For this reasons, we will be working with spanning forests (i.e., a forest that is a spanning tree in each connected component). The definition of load is the same for spanning forests.

Lemma 2 *Every k-outerplanar graph G with maximum degree at most 3 has a spanning forest of load at most $2k$.*

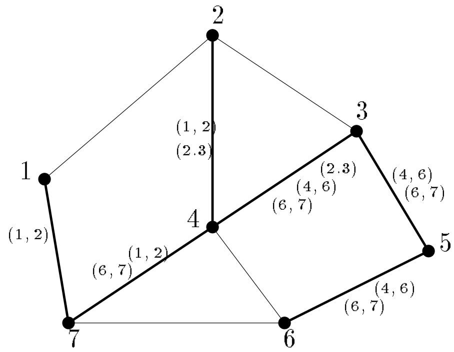

Figure 3: A spanning tree with load 3. We indicate the non-tree edges that contribute to the load of each tree-edge at the tree-edge.

We prove this lemma only if G is connected; if G is not connected then we can obtain such a spanning forest by combining the spanning trees for all connected components; this does not change the load.

So assume from now on that G is connected. We proceed by induction on k. If $k = 1$, then G is outerplanar. Since G has maximum degree 3, no vertex v can be incident to more than one interior edge, because if v is incident to any interior edge, then it must be incident to at least two vertices on the outer-face. Therefore the interior edges form a matching, and in particular a forest. Let T be a spanning tree that contains all interior edges, as well as enough edges on the outer-face to make it connected. We claim that this tree has load at most 2. Since we will use this result again in the induction step, we state it as a separate claim.

Claim 1 *Let G be a planar graph with a fixed planar embedding, and let T be a spanning tree of G such that all non-tree edges are on the outer-face of G. Then T has load 2.*

Proof: First observe that every interior face F of G can contain at most one non-tree edge. For if it contained two non-tree edges, then F would have at least two edges (u, w) and (x, y) in common with the outer-face ($w = x$ is possible). In consequence, removing edges (u, w) and (x, y) from the graph would disconnect the graph. Since T is a spanning tree, it therefore must contain one of these two edges, which contradicts that they are non-tree edges. See also Figure 4.

Now let (v, w) be a non-tree edge, and let F be the interior face incident to (v, w). (There must be such a face; otherwise (v, w) would be a bridge of the graph and any bridge must be in a spanning tree.) Consider the simple path from v to w along the boundary of F that doesn't use (v, w). This path cannot contain a non-tree edge (for (v, w) is the only non-tree edge on F), and hence is a path from x to y in T. Since there is only one such path, edge (v, w) contributes at most one to the load of every tree-edge on F.

Now consider an arbitrary tree-edge e. It is incident to at most two interior faces. Each interior face is incident to at most one non-tree edge, so each interior face incident to e contributes at most one to the load of e. So the load of e is at most two as desired. $\square$

Using this claim, we are done with the base case $k = 1$, since we found a spanning tree that has load at most $2 = 2k$.

Now for the induction step. Let G be a k-outerplanar graph, $k \geq 2$, and let G' be the graph that results from G by deleting all edges on the outer-face. G' need not be connected. We claim that G' is $(k-1)$-outerplanar. Note that if v is a vertex on the outer-face of G, then as before v

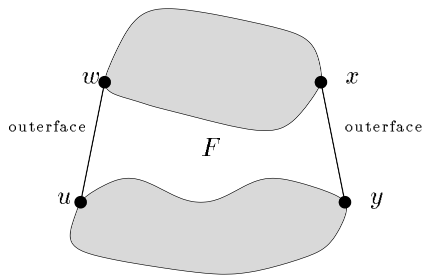

Figure 4: Two non-tree edges on one interior face would cut the graph into two disjoint parts.

is incident to at most one interior edge of G, and therefore has degree zero or one in G'. Thus, G' consists of a $(k-1)$-outerplanar graph (the graph that would have resulted from deleting all *vertices* (as opposed to edges) from the outer-face of G) plus a collection of vertices of degree zero or one. One can easily see that such a graph is $(k-1)$-outerplanar.

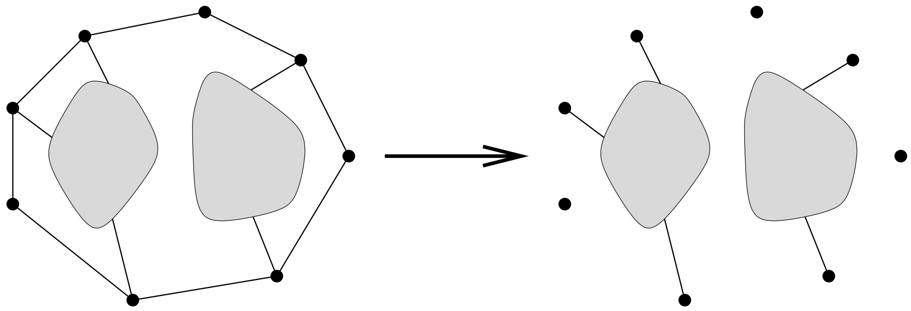

Figure 5: Obtaining G' from G.

By induction, G' contains a spanning forest T' that has load at most $2(k-1) = 2k-2$. We will now show how to expand T' into a spanning tree T of G that has at most 2 more units of load.

Let H be the graph formed by taking all edges in T' and adding the edges on the outer-face of G to it. Since G was connected and T' is a spanning forest, H is also connected. Let T be a spanning tree of H that contains all edges of T', i.e., add edges on the outer-face of G to T' until it is connected.

Notice that T is a spanning tree of graph H for which all non-tree edges are on the outer-face. By Claim 1, therefore the load of T (with respect to the non-tree edges in H) is at most 2. But T contains T', and all edges in T' had load (with respect to the edges in G') at most $2k-2$. Therefore, all edges in T have load at most $2 + 2k - 2 = 2k$ with respect to the edges in $G = G' \cup H$, which proves the claim, and therefore Lemma 2 and the second step.

2.3 From load to treewidth

The last step is to convert a spanning tree with small load into a tree decomposition of a graph. More precisely, we have the following result:

Lemma 3 *Let G be a graph with maximum degree three, and let T be a spanning tree of G that has load ℓ. Then G has treewidth at most $2\ell + 2$.*

The proof of this lemma (which also completes the proof of Theorem 1) was left as a homework.

3 Consequences

We have just shown that every k-outerplanar graph has treewidth at most $4k+2$. Recall that in a previous lecture we showed that many problems that are NP-hard for arbitrary graphs become polynomial on partial k-trees. More precisely, for problems such as Independent Set, Hamiltonian Cycle, Vertex Cover and others, there exists an algorithm to find the best solution in $O(f(k)n)$ time on a partial k-tree with n vertices. Here, $f(k)$ is some function (usually a very large function, such as 2^k or $k!$) that does not depend on n. With today's result, we have shown that all these problems are also solvable on k-outerplanar graphs, with a running time of $O(f(4k)n)$ (or actually $O(f(3k)n)$.)

Historically, this result was known before. Baker [Bak94] gave an explicit description of dynamic programming algorithms for independent set (and other problems) in k-outerplanar graphs that take time $O(8^k n)$. However, these algorithms are no easier to understand than the linear-time algorithms for partial k-trees, and also are not substantially faster since the time complexity for independent set is $O(2^k n)$ for partial k-trees, and hence $O(2^{3k-1}n) = O(8^k n)$ for k-outerplanar graphs.

References

[Bak94] B. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the Association for Computing Machinery*, 41(1):153–180, 1994.

[Bod88] Hans L. Bodlaender. Planar graphs with bounded treewidth. Technical Report RUU-CS-88-14, Rijksuniversiteit Utrecht, 1988.