Sublinear Time and Space Algorithms 2024A – Lecture 10
Sublinear-Time Algorithms for Vertex Cover in Planar Graphs*

Robert Krauthgamer

1 Vertex Cover in Planar Graphs via Local Partitioning

Problem definition:

Input: A graph $G = (V, E)$ on n vertices. We shall assume G is planar, has maximum degree $\leq d$, and is represented using adjacency lists.

Definition: A vertex-cover is a subset $V' \subset V$ that is incident to every edge.

Goal: Estimate $\mathrm{VC}(G)$ = the minimum size of a vertex-cover of G.

Theorem 1 [Hassidim, Kelner, Nguyen and Onak, 2009]: There is a randomized algorithm that, given $\varepsilon > 0$ and a planar graph G with maximum degree $\leq d$, estimates whp $\mathrm{VC}(G)$ within additive εn and runs in time $T(\varepsilon, d)$ (independent of n).

Main idea: Fix "implicitly" some near-optimal solution. Then estimate it's size by sampling $s = O(1/\varepsilon^2)$ random vertices and checking whether they belong to that solution.

Initial analysis: Let SOL be the implicit solution computed by the algorithm, let X_i for $i = 1, \ldots, s = O(1/\varepsilon^2)$ be an indicator for whether the i-th chosen vertex belongs to SOL. The algorithm outputs $\frac{n}{s}\sum_i X_i$. We will need to prove:

$$|\mathrm{SOL} - \mathrm{VC}(G)| \leq \varepsilon n$$
$$\Pr[|\tfrac{n}{s}\textstyle\sum_i X_i - \mathrm{SOL}| \leq \varepsilon n] \geq 0.9$$

The last inequality follows immediately from Chebychev's inequality, since each $X_i = 1$ independently with probability SOL/n.

Definition: We represent a partition of the graph vertices as $P : V \to 2^V$. It is called an (ε, k)-*partition* if every part $P(v)$ has size at most k, and at most $\varepsilon|V|$ edges go across between different parts.

Theorem 2: For every $\varepsilon, d > 0$ there is a polynomial $k^* = k^*(\varepsilon, d)$ such that every planar G with max-degree $\leq d$ admits an (ε, k^*)-partition.

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. The exercises are for self-practice and need not be handed in. In the interest of brevity, most references and credits were omitted.

It is proved using the famous Planar Separator Theorem (which we will not prove).

Planar Separator Theorem [Lipton and Tarjan, 1979]: In every planar graph $G = (V, E)$ there is a set S of $O(\sqrt{|V|})$ vertices such that in $G \setminus S$, every connected component has size at most $n/2$.

Remark: It extends to excluded-minor families.

Exer: Prove Theorem 3 by using the planar separator theorem recursively. What k^* do you get?

Our sublinear algorithm will not compute this partition directly (even through the Planar Separator Theorem is algorithmic), and instead it will use a "local" algorithm to compute another partition (with somewhat worse parameters).

Proof Plan for Theorem 1: Given an (ε, k)-partition P of G, we define the solution SOL by taking some optimal solution in each part of P, and adding one endpoint for each cross-edge. The following lemma is immediate.

Lemma 1a: $\mathrm{VC}(G) \leq \mathrm{SOL} \leq \mathrm{VC}(G) + \varepsilon n$.

Proof: Since $\mathrm{VC}(\cdot)$ is monotone in adding edges,

$$\mathrm{SOL} \leq \mathrm{VC}(G \setminus \mathrm{cross}(P)) + \epsilon n \leq \mathrm{VC}(G) + \epsilon n.$$

The remaining (and main) challenge is to design an algorithm that can compute $P(v)$ for a queried vertex $v \in V$ in constant time. This is called a *partition oracle*.

Note: P could be random, but should be "globally consistent" for the different queries v.

Definition: An (ε', k')-*isolated neighborhood* of $v \in V$ is a set $S \subset V$ that contains v, has size $|S| \leq k'$, the subgraph induced on S is connected, and the number of edges leaving S is $\mathrm{e_{out}}(S) \leq \varepsilon' |S|$.

Algorithm Partition (used later as oracle):

Remark: It uses parameters ε', k' that will be set later (in the proof)

1. $P = \emptyset$
2. iterate over the vertices in a random order $\pi_1, \ldots, \pi_n$
3. if π_i is still in the graph then
4. if π_i has an (ε', k')-isolated neighborhood in the current graph
5. then $S =$ this neighborhood
6. else $S = \{\pi_i\}$
7. add $\{S\}$ to P and remove S from the graph
8. output P

Lemma 1b: Fix $\varepsilon' > 0$. Then a random vertex in G has probability at least $1 - 2\varepsilon'$ to have a $(k^*(\varepsilon'^2, d), \varepsilon')$-isolated neighborhood.

Proof of Lemma 1b: Was seen in class, by considering the $(\varepsilon'^2, k^*(\varepsilon'^2, d))$-partition guaranteed to exist by Theorem 2.

We will continue next class, and show that the above algorithm indeed provides a partition oracle.