CS173 Lecture B, November 5, 2015

Tandy Warnow

November 2, 2015

CS 173, Lecture B
November 5, 2015
Tandy Warnow

Today

- NP-hard problems: how to deal with them
- Approximation algorithms with guaranteed error bounds
- A 2-approximation algorithm for Minimum Vertex Cover

Minimum Vertex Cover

A **vertex cover** for graph $G = (V, E)$ is a subset $V_0 \subseteq V$ such that every edge in G has at least one of its endpoints in V_0.

A **minimum vertex cover** for G is a vertex cover V_0 such that $|V_0| \leq |V'|$ for all vertex covers V' of G.

Finding the minimum vertex cover for a graph is an NP-hard problem.

Exhaustive search for Minimum Vertex Cover

To find an optimal (i.e., smallest) vertex cover for a graph, we can use exhaustive search.

If G contains no edges, then return $\emptyset$. Otherwise, let $G = (V, E)$ have n vertices. DO:

- For $k = 1$ up to n
  - Examine each of the subsets of k vertices from V to see if it is a vertex cover. If any is, then return that subset.

The running time for this algorithm is $\Theta(n^{K+1})$ where K is the size of the minimum vertex cover. This is not polynomial in the input size (since K need not be a constant).

Can we do better?

Dealing with NP-hard problems

Unless $P = NP$, we won't be able to solve any NP-hard problem in polynomial time (not on all inputs, anyway).

We can try heuristic searches (hill-climbing, sometimes using randomization to get out of local optima, etc.) but

- There are no guarantees of optimal solutions.
- They don't necessarily run in polynomial time.
- Sometimes the solutions they find are very poor (far from optimal).

Approximation algorithms

One type of approach is to find an *approximate* solution, which means that we are guaranteed to get a *feasible solution* (i.e., a valid output) which is not too far from optimal.

For example, a c-approximation algorithm for Minimum Vertex Cover would return a vertex cover V_0 such that

$$|V_0| \leq c|V_{opt}|,$$

where V_{opt} is an optimal solution to Minimum Vertex Cover.

The smaller that c is the better the approximation ratio.

We will show an easy polynomial time algorithm that produces a 2-approximate solution to Minimum Vertex Cover!

Maximal Matching

A matching is a subset M of the edges of a graph such that no two edges in M share an endpoint.

M is a **maximal matching** of G if

- M is a matching for G and
- $\forall M'$ if $M \subset M' \subseteq E$ then M' is not a matching

M is a **maximum matching** of G if

- M is a matching of G and
- $\forall M'$ if M' is a matching of G then $|M| \geq |M'|$.

For an example of a maximal but not maximum matching, consider the graph P_4 with four vertices, a, b, c, d, and three edges $(a, b), (b, c), (c, d)$. The middle edge (b, c) is a maximal matching but not a maximum matching.

Two helpful lemmas

- Lemma 1: If $G = (V, E)$ is a graph and $M \subseteq E$ is a matching, then every vertex cover for G has at least $|M|$ vertices.
- Lemma 2: Let $G = (V, E)$ be a graph, $M \subseteq E$ be a *maximal* matching, and V_0 be the set of endpoints of edges in M. Then V_0 is a vertex cover for G.

Proof of Lemma 1

Lemma 1: If $G = (V, E)$ is a graph and $M \subseteq E$ is a matching, then every vertex cover for G has at least $|M|$ vertices.

Proof: Let X be a vertex cover for G.

- Every edge in M has to have at least one of its endpoints in X, by the definition of vertex cover.
- Since no two edges in M share any endpoints, each edge contributes a different vertex to X.
- Hence, $|X| \geq |M|$.

q.e.d.

Proof of Lemma 2

Lemma 2: Let $G = (V, E)$ be a graph, $M \subseteq E$ be a *maximal* matching, and V_0 be the set of endpoints of edges in M. Then V_0 is a vertex cover for G.
Proof: By contradiction. Suppose for some graph G and some maximal matching M of G, the set V_0 of endpoints of edges in M is not a vertex cover.

Then there is some pair of vertices x, y in G, neither of which is an endpoint of any edge in M, and that are adjacent (so $(x, y) \in E$).

Consider $M' = M \cup \{(x, y)\}$.
M" is a matching of G and $M \subset M'$.
Hence M is not a maximal matching, contradicting our hypothesis.
q.e.d.

Another theorem

Theorem: Let G be any finite simple graph, let M be a maximal matching for graph G, and let V_0 be the set of endpoints of edges in M. Let V_1 be the smallest vertex cover for G. Then $|V_0| \leq 2|V_1|$.

Proof

Theorem: Let G be any finite simple graph, let M be a maximal matching for graph G, and let V_0 be set of endpoints of edges in M. Let V_1 be the smallest vertex cover for G. Then $|V_0|$ is a vertex cover for G, and $V_0 \leq 2|V_1|$.

Proof: By Lemma 1, $|V_1| \geq |M|$, and equivalently $|M| \leq |V_1|$.
By Lemma 2, V_0 is a vertex cover for G.
Since $|V_0| = 2|M|$, it follows that $|V_0| \leq 2|V_1|$.
q.e.d.

Approximation Algorithm for Vertex Cover

Input: $G = (V, E)$
Output: Vertex cover V_0 of G such that V_0 contains no more than twice the number of vertices in a minimum vertex cover

Algorithm:

- Greedily construct a maximal matching M
- Let V_0 be the set of endpoints of M

The greedy algorithm to construct a maximal matching:

- Let $M \leftarrow \emptyset$
- While G has an edge (a, b), DO:
  - Set $M \leftarrow M \cup \{(a, b)\}$
  - Delete a, b from G and their incident edges
- Return M

The running time is easily seen as polynomial.

Vertex Covers and Independent Sets

A **vertex cover** for graph $G = (V, E)$ is a subset $V_0 \subseteq V$ such that every edge in G has at least one of its endpoints in V_0.

An **independent set** for graph $G = (V, E)$ is a subset of the vertices V_0 so that no two vertices in V_0 are adjacent (i.e., for all $\{v, w\} \subseteq V_0, (v, w) \notin E$).

Theorem: V_0 is a vertex cover in graph $G = (V, E)$ if and only if $V - V_0$ is an independent set of G.
Hence, finding a minimum vertex cover is equivalent to finding a maximum independent set.
How does the 2-approximation algorithm for minimum VC help you find an approximate solution to maximum independent set?

What does this tell you?

Suppose I have a graph G with 100 vertices, and I run the 2-approximation algorithm for Minimum Vertex Cover and get a set V_0 of 50 vertices.

- What can I say about the size of a minimum vertex cover?
- What can I say about the size of a maximum independent set?

What does this tell you?

Suppose I have a graph G with 100 vertices, and I run the 2-approximation algorithm for Minimum Vertex Cover and get a set V_0 of 80 vertices.

- What can I say about the size of a minimum vertex cover?
- What can I say about the size of a maximum independent set?

What does this tell you?

Suppose I have a graph G with 100 vertices, and I run the 2-approximation algorithm for Minimum Vertex Cover and get a set V_0 of 90 vertices.

- What can I say about the size of a minimum vertex cover?
- What can I say about the size of a maximum independent set?