Title

Computer Science 604

Advanced Algorithms

Lecture 9

David Juedes

School of EECS

juedes@cs.ohiou.edu

Examples

Consider the minimization problem

MIN Vertex Cover

Given: a graph $G = (V, E)$.

Find: $C \subseteq V$ such that for all $(u, v) \in E$, either $u \in C$ or $v \in C$ and $|C|$ is the minimal.

Remark: It is well-known that the decision problem

Vertex Cover: given a graph G and an integer k, determine whether G has a vertex cover of size k

is NP-complete.

An approximation algorithm for VC

An approximation algorithm for MIN Vertex cover is based on algorithms for maximal matching.

A *matching* in a graph $G = (V, E)$ is a set $E' \subseteq E$ such that no two edges in E' share a common endpoint.

A matching $E' \subseteq E$ is *maximal* if every remaining edge in $E - E'$ shares a common endpoint with an edge in E'.

Maximal Matchings

It is easy to construct a polynomial-time algorithm to produce a maximal matching.

How do you do this?

An algorithm for Vertex Cover

(i) Set $E'' = E$; $C = \emptyset$

(ii) Pick an edge $(u, v) \in E''$, put u and v in C

(iii) Put $(u, v) \in E'$;

(iv) Eliminate all edges in E'' that are incident upon u or v

(v) Repeat (ii), (iii), and (iv) until $E'' = \emptyset$

Is it clear that this algorithm runs in polynomial time?

Analysis

What do we know about E'?

Is C a vertex cover? Why?

How close is it to optimal?

MIN Vertex Cover

The performance of the approximation algorithm for MIN Vertex Cover

Let I be an instance of the Vertex cover problem. Then, our approximation algorithm achieves

$$A(I) \leq 2 * OPT(I).$$

Why?

Notice that our algorithm constructs a maximal matching E'.

If we examine a vertex cover V', then at least one vertex from each edge in E' must be in V'. Thus, $|V'| \geq |E'|$.

Maximal Matching vs. VC

Now, the vertex cover constructed in our algorithm has size $2 \cdot |E'|$. Hence, the size of our vertex cover is at most $2 * OPT(I)$.

Is our analysis optimal???

Can you give an example of a graph where

1. Our approximation algorithm produces a solution of size n, and

2. The optimal solution is of size $n/2$?

Conclusion

Hence, the approximation ratio for this algorithm is exactly 2.