College of Computer Science
Northeastern University
COM 3391: Advanced Algorithms

Lecture 4, part 2
20 October 1999
Scribe: Prakash Viswanathan

Approximation Algorithms for Vertex Cover

The material covered in this lecture are based on the Chapter 4 of the lecture notes of Rajeev Motwani [Mot92].

1 Introduction

This lecture looks at some known algorithms for the *vertex cover*, which is known to be *NP-complete* and, compares their approximation of the optimal.

Definition 1 (Vertex Cover) *A* vertex cover *in an undirected graph* $G = (V, E)$ *is the set of vertices* C *such that each edge of* G *has at least one endpoint in* C*.*

- **Instance:** $G = (V, E)$.
- **Solutions:** *Set of vertices that cover every edge.*
- **Value:** $\mid V' \mid$, *the number of vertices in the vertex cover* V' *and, the goal is to minimize it.* ■

2 Approximating Vertex Cover

2.1 Greedy Algorithm One

The most natural heuristic is a greedy algorithm which repeatedly picks an edge that has not yet been covered, and places one of its endpoints in the current covering set.

Input: Unweighted graph $G = (V, E)$
Output: Vertex cover C.

1. $C \leftarrow \phi$
2. $G'(V', E') = G(V, E)$
3. Repeat while there are edges in G'
   - Pick $(u, v) \in E'$
   - $C = C \cup \{u\}$
   - $V' = V' - \{u\}$
   - $E' \leftarrow E' - \{e \in E' | e \text{ is incident on } u\}$ ■

Claim 1 *Above algorithm does not achieve any bounded ratio.*

Proof:

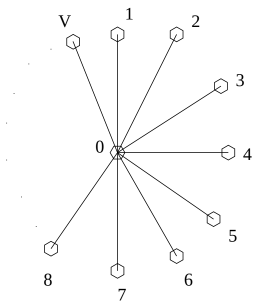

Figure 1: Bad case for Greedy Algorithm One.

Consider a graph as shown in Figure 1. The algorithm might consider edges $(0,1),(0,2),\ldots,(0,V)$ in order and pick the larger numbered vertex in each edge, giving it a vertex cover of size $\mid V \mid$, whereas, the optimal would select only vertex 0. It follows that the ratio achieved by this algorithm is no better than $\mid V \mid$.

■

2.2 Greedy Algorithm Two

One obvious strategy of achieving a better ratio is that of modifying Greedy Algorithm One to be less arbitrary in its choice of vertices to be included in the cover. A natural modification is to repeatedly choose vertices which are incident to the largest number of *currently* uncovered edges.

Input: Unweighted graph $G = (V, E)$
Output: Vertex cover C.

1. $C \leftarrow \phi$
2. $G'(V', E') = G(V, E)$
3. Repeat while there are edges in G'
   - Pick vertex v in G' with maximum degree
   - $C = C \cup \{v\}$
   - $V' = V' - \{v\}$
   - $E' \leftarrow E' - \{(u, v) \in E'\}$

■

Claim 2 *Greedy Algorithm Two does not achieve any bounded ratio.*

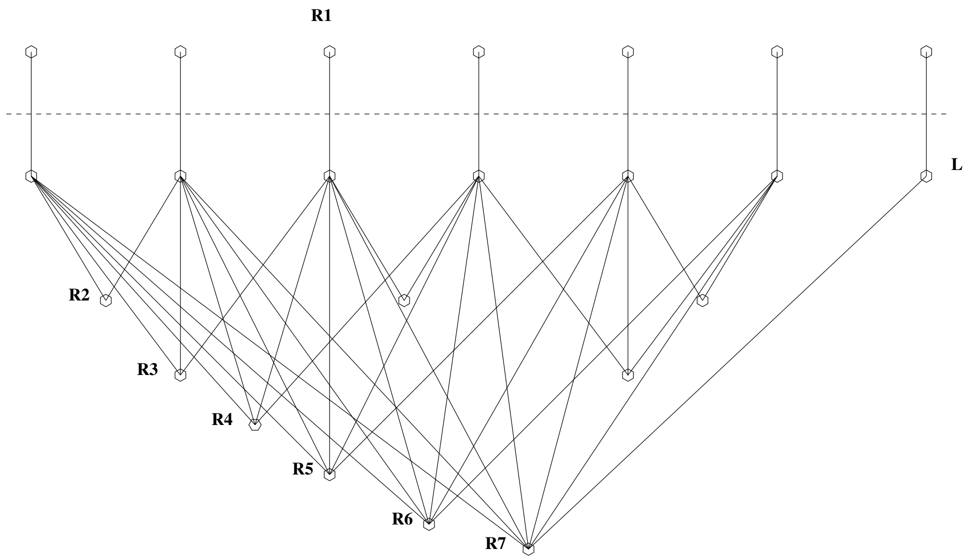

Figure 2: The graph $B(L, R, E)$ with $r = 7$.

Proof:

Consider the bipartite graph $B(L, R, E)$ depicted in Figure 2.2. The vertex set L consists of r vertices. The vertex set R is further sub-divided into r sets called R_1, R_2,..., R_r. Each vertex in R_i has an edge to i vertices in L and no two vertices in R_i have a common neighbor in L; thus, $\mid R_i \mid = \lfloor r/i \rfloor$. (It is possible that not all vertices in L have a neighbor in a particular R_i.) It follows that each vertex in L has degree at most r and each vertex in R_i has degree i. The total number of vertices n is $\Theta(r \log r)$.

Consider the behavior of the greedy algorithm on the graph B. Suppose that (out of sheer bad luck) the algorithm picks vertices from R_r at the very first stage. Then it picks vertices from R_{r-1}. In general, it picks the highest degree vertices from R at each stage. Therefore the vertex cover chosen is $C = R$.

Now, $\Sigma R_i = L(1 + 1/2 + 1/3 + \ldots + 1/L) - L$

From Figure 2.2,
$\int_1^L dx/x \leq (1 + 1/2 + 1/3 + \ldots + 1/L) \geq \int_0^{L-1} dx/x$
$\Longrightarrow (1 + 1/2 + 1/3 + \ldots + 1/L) = \Theta(\ln L)$

Therefore $\Sigma R_i \approx L\Theta(\ln L) - L = L\Theta(\ln L)$

But L is itself a vertex cover since the graph is bipartite. It follows that the ratio achieved by this algorithm is no better than $\mid R \mid / \mid L \mid = \Omega(\ln n)$. ■

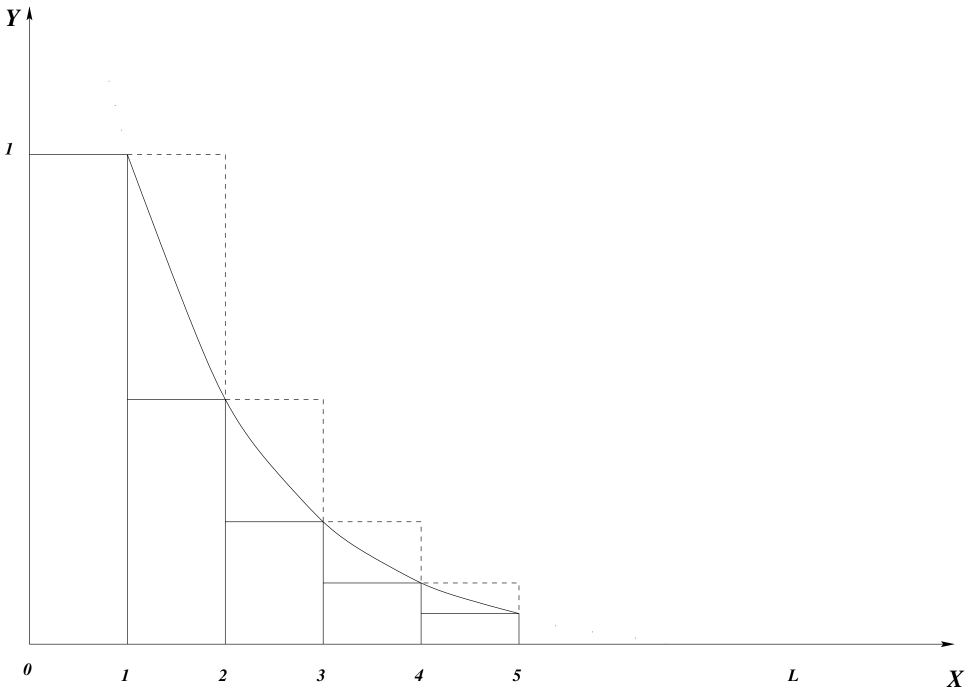

Figure 3: Calculation of the series $(1 + 1/2 + 1/3 + \ldots + 1/L)$

2.3 Greedy Algorithm Three

We now describe a different heuristic which achieves a bounded ratio for the vertex cover problem. The basic idea is to modify Greedy Algorithm One by placing *both* its end-points of some uncovered edge into C.

Input: Unweighted graph $G = (V, E)$
Output: Vertex cover C.

1. $C \leftarrow \phi$
2. $G'(V', E') = G(V, E)$
3. Repeat while there are edges in G'
   - Pick $(u, v) \in E'$
   - $C = C \cup \{u\} \cup \{v\}$
   - $V' = V' - \{u, v\}$
   - $E' \leftarrow E' - \{e \in E' | e \text{ is incident on } u \text{ or on } v\}$ ■

Definition 2 (Matching) *A* matching *in a graph is a subset of edges in which no two edges share a vertex.*

Definition 3 (Maximal Matching) *A* maximal matching *is a matching to which no edge in the graph can be added. That is, a matching is maximal if it is not contained in any larger matching.*

Theorem 1 *If M is a matching in G, then the minimum vertex cover is at least $\mid M \mid$.*

Proof: Consider any edge in the matching. The vertex cover C, must include at least one of the two vertices, because if it does not have either of the vertices, then there is one edge in the graph that is not covered, contradicting the claim that C is a vertex cover. ■

Greedy Algorithm Three actually picks the maximal matching in G.

Theorem 2 *Greedy Algorithm Three always computes a vertex cover C in the input graph G that is a 2-approximation of the optimal.*

Proof: To see that the ratio is 2, consider the edges in M, the maximal matching in G. To cover these edges we need at least $\mid M \mid$ vertices, since no two of them share a vertex. This implies that the optimal vertex cover has size at least $\mid M \mid$. The cover C contains exactly $2 \mid M \mid$ vertices. Hence the theorem.

■

References

[Mot92] R. Motwani. Lecture notes on approximation algorithms: Volume I. Technical Report STAN-CS-92-1435, Department of Computer Science, Stanford University, 1992. Postscript document is available at http://Theory.Stanford.EDU/˜rajeev/postscripts/approximations.ps.gz (134 pages).