1 Set-01-Problem-02, Text-P8-1.4

Perhaps the first strategy one tries when designing an algorithm for an optimization problem is the greedy strategy. For the vertex cover problem, this would involve iteratively picking a maximum degree vertex and removing it, together with edges incident at it, until there are no edges left. Show that this algorithm achieves an approximation guarantee of $O(\log n)$. Give a tight example for this algorithm.

Proof: First, we prove that the greedy algorithm achieves an approximation guarantee of $O(\log n)$. Consider graph $G = (V, E)$. Let d_i denote the degree of the vertex that the greedy algorithm picks in step i, and let ℓ denote the total number of execution steps of an instance of the greedy algorithm. A cost of $c(e_{ij}) = 1/d_i$ is assigned to every edge e_{ij} incident to the removed vertex in step i. Note that d_i refers to the degrees in their current reduced graph. Also, let $|VC|$ and $|OPT|$ denote the vertex cover the greedy algorithm and the optimal respectively. We have

$$|VC| = \sum_{i=1}^{i=\ell} d_i \frac{1}{d_i} = \sum_{e \in E} c(e)$$

Since we are picking maximum degree vertex in each round, we have

$$c(e_{ij}) = \frac{1}{d_i} \leq \frac{|OPT|}{|E| - d_1 - d_2 - \ldots - d_{i-1}},$$

which consequently shows that

$$|VC| \leq |OPT| H_{|E|} = O(\log n^2)|OPT| = O(\log n)|OPT|.$$

This completes the proof.

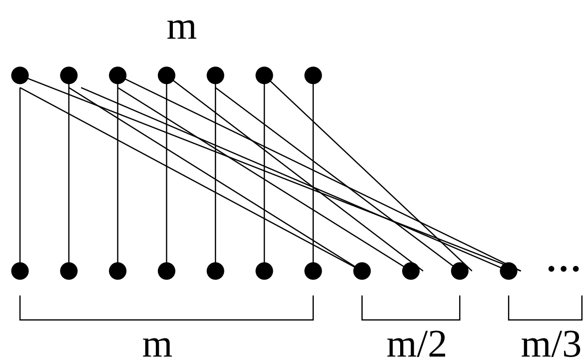

Figure 1: A tight bound on the performance guarantee of the greedy vertex cover.

Consider Figure 1. A sequence of m vertices are placed on the top, and a sequence of $m + \lfloor m/2 \rfloor + \lfloor m/3 \rfloor + \ldots + m/m$ vertices are at the bottom. Each of the first m vertices at the bottom have an edge with one vertex on the top, and each vertex on top have only one edge with the m vertices at the bottom; each of the next $\lfloor m/2 \rfloor$ vertices have 2 edges with the vertex on the top, and each vertex on top have only one edge with the $\lfloor m/2 \rfloor$ vertices at the bottom; and so on. In this graph, we know that, at the bottom, there are m/m vertices with degree m; there are $\lfloor m/(m-1) \rfloor$ vertices with degree $\lfloor m/(m-1) \rfloor$, and so on. Vertices on the top all have degree m at the beginning. During the greedy algorithm execution, we assume at this moment that it always pick vertices at the bottom starting from the right side, since the bottom vertices always have no less degree than the vertices on the top. Thus, the greedy algorithm would have a vertex cover of size $n - m$, where n is the total number of vertices. While it is easy to see that all the vertices on the top is one vertex cover, and the size is m.

We now show that the approximation ratio is at least $O(\log n)$. From our construction and the algorithm, we know that the approximation

$$\frac{|VC|}{|OPT|} \geq \frac{n-m}{m} = \frac{n}{m} - 1 \geq \frac{\sum_{1 \leq i \leq m} \frac{m}{i}}{m} - 1 \geq \ln m - 1 \geq \frac{1}{2} \ln m,$$

since $\lfloor r \rfloor \geq r - 1$ for any $r \in R^+$. We know that $n \leq m + \sum_{1 \leq i \leq m} \frac{m}{i} \leq 2m \ln m$ (since $\lfloor r \rfloor \leq r$ for any $r \in R^+$), thus $\ln n \leq \ln 2 + \ln m + \ln \ln m \leq 2 \ln m$. This proves that the example does place an lower bound of $\Theta(\log n)$ on the greedy vertex cover algorithm.

Note that there are tie-breakings during the algorithm, one way to remove tie-breaking is to remove the first m vertices at the bottom.

■