

Wrong greedy for minimum vertex cover

Find an example of (family of) graphs for which the following greedy approach fails to give a 2-approximation for the minimum vertex cover problem (and prove why this is so). Start out with an empty \hat{S} . Choose each time a vertex v with the largest number of incident edges in the current graph. Add v to \hat{S} and remove its incident edges. Repeat the process on the resulting graph as long as there are edges in it. Return $|\hat{S}|$ as the approximation of the minimal size of a vertex cover for the original input graph.

Generalize your argument to show that the above greedy algorithm cannot actually provide an r -approximation for any given constant $r > 1$.

Solution

We will use a partitioned graph $G = (V, E)$. The set of vertices V is divided in two partitions A and B . The graph is such that all the edges connect two vertices one from A and one from B (i.e. we don't allow intra-partition edges).

In this way we can see that the degree of A and B are the same (we define the degree of a set of vertices to be the sum of the degrees of the vertices), they are exactly $|E|$. We name $n = |B|$ and $m = |A|$. Note that if the graph is connected, one minimum vertex cover is exactly A or B depending on which is the minimum between m and n .

We will build the graph in a way such that the minimum vertex cover is B (i.e. $m > n$), but the greedy algorithm can choose all the nodes of A as minimum vertex cover, and we will show that $m > 2n$ if n is sufficiently large.

The idea is to fix n and then build the graph having m depending somehow on n , such that the m node of A can be chosen by the algorithm.

The construction works this way:

1. set $i = 1$, $A = \{\}$
2. create $\lfloor n/i \rfloor$ new nodes with degree i
3. insert the nodes obtained in A
4. connect each edge of those nodes to a *different* node in B following a round robin policy ¹
5. increase i , if it is not greater than n then go to 1. else you are done

Following this construction the following facts are true:

fact 1 At the i -th iteration we add a total number of new edges *less or equal* than n , each one connected to a different node in B .

fact 2 Since we distribute the edges in a uniform way on the nodes in B , the degree of a node in B at iteration i will be less or equal than i (i.e. for each iteration we add zero or one edge to each node in B)

fact 3 A node created at iteration i has exactly degree i .

Now let's study what happens when we run the greedy algorithm on the graph obtained in the proposed way.

At the very beginning this algorithm will select a node between one of the nodes in B (each of them has degree less or equal than n) and the last node inserted in A , which has degree n exactly. If the greedy algorithm select this last node from A the algorithm will remove the edges of this node (and the node itself) and so each node in B will have its degree decreased by 1. The remaining graph will be the one present during the $(n - 1)$ -th iteration of the construction!

¹this round robin policy is to be considered over the whole execution of the algorithm, i.e. at iteration i we will start connecting the new edges to the node in B "immediately after" the last one connected at the $i - 1$ iteration

By construction we know that at iteration $n - 1$ we have $\lfloor n/n - 1 \rfloor$ nodes in A each of which with degree $n - 1$. From **fact2** we also know that at the $n - 1$ -th iteration the maximum degree of the nodes in B is *at most* $n - 1$.

We are in the same situation of the very beginning and the greedy algorithm can choose again a node in A .

With the same reasoning of above we can observe that at any iteration of the greedy algorithm the graph will have a node in A which has degree greater or equal than the degree of each node in B .²

Therefore the greedy algorithm running on a graph constructed as described above can return $m = |A|$ as approximation of the MVC of that graph, i.e. there exists a sequence of choices that will select all the nodes in A . If $m > 2n$ the result will not be a 2-approximation of MVC. We can see that this happens with $n \geq 6$

Infact we can see that with our construction A will have $m = 6 + \frac{6}{2} + \frac{6}{3} + \lfloor \frac{6}{4} \rfloor + \lfloor \frac{6}{5} \rfloor + 1 = 14$ nodes and so the greedy algorithm can return 14 that is not a 2 approximation of the MVC problem.

GENERALIZATION

We can generalize the solution given by the greedy algorithm running on our constructed graph so that we can see that for any $r > 1$ the algorithm can provide a non valid r -approximation.

We want to show that for any $r > 1$ there exist n such that the greedy algorithm will not provide an r -approximation.

Using our graph construction method, given n as the correct minimal size of the MVC, the number of nodes in A will be

$$m = \sum_{i=1}^n \lfloor n/i \rfloor$$

so we want to show that

$$\sum_{i=1}^n \lfloor n/i \rfloor > rn$$

We can observe that

$$\sum_{i=1}^n \lfloor \frac{n}{i} \rfloor > \sum_{i=1}^n (\frac{n}{i} - 1) = n \sum_{i=1}^n \frac{1}{i} - n = n(-1 + \sum_{i=1}^n \frac{1}{i}) \approx n \log n$$

Since $n \log n$ grows faster than rn , given $r > 1$ there will always be a value of n that will cause the greedy algorithm to provide a value $m = \sum_{i=1}^n \lfloor n/i \rfloor > rn$ that is not an r -approximation.

²of course the greedy algorithm could select a note in B as well, but this will not affect our argument because we are trying to prove that there is a sequence of possible wrong choices