Wrong greedy for minimum vertex cove

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 $\tilde{S}$. Choose each time a vertex v with the largest number of incident edges in the current graph. Add v to $\tilde{S}$ and remove its incident edges. Repeat the process on the resulting graph as long as there are edges in it. Return $|\tilde{S}|$ 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).

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 choose n, then build the graph having m depending somehow on n, such that the m node of A can be chosen by the algorithm. Consider a graph with $k = |E|$ edges, the best way to build B is to distribute equally the edges on the nodes, this way the maximum degree based on which we will choose the node to take will be minimized. For A instead we need to distribute the edges in a very non-balanced way, to obtain a grater max.

The construction work this way:

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

Note that for any i you generate a total number of edges minor or equal to n. Since you distribute the edges in a uniform way on B you know that the degree of a node in B at iteration i will be less or equal to i (for each integer from 1 to i we add zero or one edge) which is the exact degree of the nodes generated at this step.

Then the greedy algorithm will have to choose at the very beginning between one of the nodes of B, with max n degree or the last node inserted in A, which has degree n. The greedy algorithm can choose the node from A, if it does, the remaining graph is the one present during the (n-1)th iteration of the construction (note that removing the node means also removing its edges and then decreasing the degree of the nodes on B).

At any iteration the greedy algorithm will have a node in A which has degree greater or equal to the remaining degree of the node on B, since for what we said before the degree of the nodes of B is less then the number of iteration i in which we inserted the node in A.