Max Wisniewski

Dozent : Panos Giannopoulos

Exercise 1

Consider the 2-approximation algorithm for the cardinality (i.e. unweighted) Vertex Cover problem that is based on computing a Maximal Matching. What approximation factor does it give for the general weighted vertex cover problem, where the vertices have arbitrary positive weights?

Solution:
Because δ is a function which domain is the set of instances for the problem we might take any information from the graph we need. The general idea to estimate the approximation factor is, that on each edge from the Maximal Matching, there might stand the maximal value of all nodes.
Because at least one of these nodes has to be part of the Minimum Vertex Cover we consider the worst case. One of these nodes contains the smallest possible value, the other one has the biggest on.
Now the weight of the two nodes on the edges we took is $\frac{max+min}{min} = \frac{max}{min} + 1$ times as large as the smallest posibility might be.
We sum up every node in the Maximal Matching and because we know we know each pair of the matching is at most $\frac{max}{min} + 1$ times as large, we can use the destribution law to extract the factor

$$\delta(G(V,E)) = \frac{\max(V)}{\min(V)} + 1.$$

Claim 1 $|C| \leq \left(\frac{\max(V)}{\min(V)} + 1\right) \cdot OPT$ and C is a vertix cover.

Proof
In lecture the Theorem was given without proof, that for a unweighted graph $\mathfrak{A}$'s returnvalue is a Vertex Cover. If the graph is weighted, this fact is not changed in any way, because the definition of Vertex Cover is independend of any weight. So this claim holds.

$|C| \leq \left(\frac{\max(V)}{\min(V)} + 1\right) \cdot OPT$ holds, because for each edge we took, we sum up a value that is δ times larger, than any in the Minimum Vertix Cover. The proof that we do not have more edges in the maximal matching then we have nodes in the Minimum Vertix Cover is the only remaining one.
But if we have less nodes in the Vertex Cover than edges the Maximal Matching takes, we have at least one edge in the Maximal Matching with both nodes not containt in the Vertex Cover. But, in case we have no multiedges, this can't be a Vertex Cover.

□

Claim 2 $\left(\frac{\max(V)}{\min(V)} + 1\right) \cdot OPT$ is a tight upper bound.

Proof

Let G_n be a family of graphs, s.t.
$G_i(V_i, E_i) = (\{a_t \mid 0 \leq t \leq 2i\}, \{\{a_t, a_{t+1 \mod 2i} \mid 0 \leq t \leq i\})$
and together with a weight function
$w_i : V_i \to Q^+, a_t \mapsto \frac{1}{2} \cdot \left((1 + (-1)^t) + (1 + (-1)^t) \cdot c\right)$ with $c \in \mathbb{Q}^+$ for each $i \in \mathbb{N}$ is a graph of this family.

For each $G_i(V_i, E_i)$ the Maximal Matching will deliver a C through the algorithm $\mathfrak{A}$ that will consist of every node ($C = V_i$). Our weight function returns only 2 values. Either 1 or c. Therefore the minimum of all nodes is 1 and the maximum is c. We have as many minimum values on hte nodes as we have maximum values and each was taken from the Maximal Matching. Therefore $obj_{VC}(G_i, C) = c \cdot i + i$ holds.

As we can easily see, the minimum vertex cover will take every other node, namly the nodes with the minimum values. With that every edge has exactly one node taken in the Minimum Vertex Cover. For this graph therefore $obj_{VC}(G_i, OPT) = i$ holds.

Now $|C| = \left(\frac{\max(V)}{\min(V)} + 1\right) \cdot OPT$ holds, because

$$\begin{aligned} |C| &= c \cdot i + i \\ &= \left(\frac{c}{1} + 1\right) \cdot i \\ &= \left(\frac{\max(V_i)}{\min(V_i)}\right) \cdot OPT \text{ holds.} \end{aligned}$$

□

Exercise 2

Consider the following greedy algorithm for carinality Vertex Cover:

```
• Input : Graph $G(V, E)$
• $C := \emptyset$
• While $C$ is not a vertex cover of $G$
    – let $u$ be the vertex incident to the most uncovered edges
    – $C := C \cup \{u\}$
• return $C$
```

Is this a constant-factor approximation algorithm?

Solution
This is not a constant-factor approximation algorithm. If we look at the figure 1 we can see, that in the first execution of the *while loop* the vertex with that is not contained in the set C an has the most uncovered edges is the one on the right with the value c. If we assume, that c is greater than two, the OPT is exactly two. The δ in our factory approximation now depends on the value of c. Therefore it is not constant but linear dependend of c.

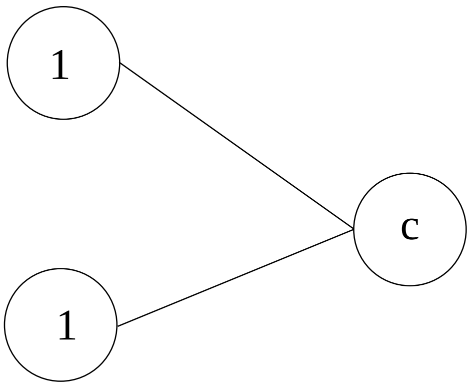

Figure 1: Counterexample for exercise 2

Exercise 3

Consider the following approximation algorithm for the cardinality Vertex Cover problem: Find a depth first search tree T in the given input graph G, and return the set C of all the nonleaf vertices of T. Show that this is also a 2-approximation algorithm.

Solution
Let $\mathfrak{D}$ be the Algorithm described above. I assume the Graph G is connected. Otherwise we can find a simple counterexample.

Claim 3 The set C returned by $\mathfrak{D}$ is a vertex cover.

Proof

Let $G(V, E)$ be a valid instance for this problem and T the DFS tree generated by $\mathfrak{D}$. A DFS tree in a Connected Graph visits all nodes. If not, there would exist an edge from a visited node to an unvisited one alongside the algorithm should have taken its path.

We now assume, there exists an edge $e = (i, j) \in E$ with $\{i, j\} \cap C = \emptyset$.
Every node, as stated above is visited, so i, j may be either nonleaf vertices or leafs of the tree. If either of them would be a nonleaf vertex it would be in C. The only left possibility is, that both i, j are leafs.
Let i be the vertex visited first from the DFS algorithm w.l.o.g.. Then j has to be a child of i, because the algorithm expects first all direct childs of i before it returns to the parent of i. Because there exist an edge from i to j this edge is either taken in the DFS or there exists an edge from one of i children to J. Either way i can't be a leaf.

One can conclude, there is no edge, that has no vertex in the set C and that implies, C is a Vertex Cover.

$\square$

Claim 4 $\mathfrak{D}$ is a $(|V| - 1)$-approximation algorithm.

Proof

A finite tree has at least one leaf. If the tree is a single Path, then at most $|V| - 1$ nodes are taken. This is an upper bound because there can't be an more nodes, that could be taken.

Claim 5 $(|V| - 1) \cdot OPT$ is a tight upper bound.

Proof

Let K_n be the family of complete graphs with n vertices.
As easy to see, we only need one node to have a Vertex Cover for this graph und therefore it is the minimum.

The algorithm $\mathfrak{D}$ will build a DFS tree out of the complete graph. Because every node has an edge to every node the resulting tree T will be a single path with all nodes on it. This path has one leaf, because it is finite on n vertices. The output of $\mathfrak{D}$ would contain exactly $n - 1$ vertices. This result is $n - 1$ times as large as the optimum.