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 one.
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 distribution 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.

$\square$

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.
$G_i(V_i, E_i) = (\{a_t \mid 0 \le t \le 2i\}, \{\{a_t, a_{t+1 \mod 2i} \mid 0 \le t \le i\})$
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

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

$\square$

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
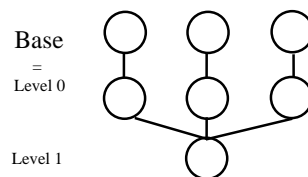   - $C := C \cup \{u\}$

- return C

Figure 1: Anchor of the graph familiy (G_1)

Is this a constant-factor approximation algorithm?

Solution

Claim 3 The algorithm is not a constant-factor approximation algorithm.

Proof

We construct a family of graphs that have a factor, dependend in the amount of vertices.
Let $G_n(V_n, E_n)$ be a family of graphs constructed as inductiv as followed:
$G_1(V_1, E_1)$ is a graph of the pairwise connected vertices and one vertex, that has a edge to one vertex of each pair. In figure 1 the graph is visualized.

Now for the Induction G_i we take three graph of G_{i-1}. Further we add a number of new vertices to the graph and connect each of these vertices with each vertex in every baselayer in the three G_{i-1} graphs, therefore the degree of these vertices will be 3^i. The constructed graph looks like the example in figure 2 for the G_2.

The number of new vertices we will add must be as large as possible, but only that much, that our 3^i vertices in the baselevel will have at most the degree $3^i - 1$. From this we construct two formulas η for the amount of vertices on level k and Ψ for all nodes (the pairs in the baselevel count as 1 per pair).

$$
\begin{aligned}
\eta \quad &: \quad \mathbb{N} \longrightarrow \mathbb{N} \\
0 \quad &\mapsto \quad 1 \\
i \quad &\mapsto \quad 3^i - \left(\sum_{k=0}^{i-1} \eta(k) \right) - 1 \quad , i > 0
\end{aligned}
$$

$$
\begin{aligned}
\Psi \quad &: \quad \mathbb{N} \longrightarrow \mathbb{N} \\
i \quad &\mapsto \quad 3^i + \sum_{k=1}^{i-1} 3^{k-1} \times \eta(i - k - 1)
\end{aligned}
$$

As one can easily see, the Vertex Cover must contain one vertex of each pair in the base level at least. Because every other edge is connected to the baselevel, the Minimum Vertex Cover will contain exactly one vertex (the one every edge is
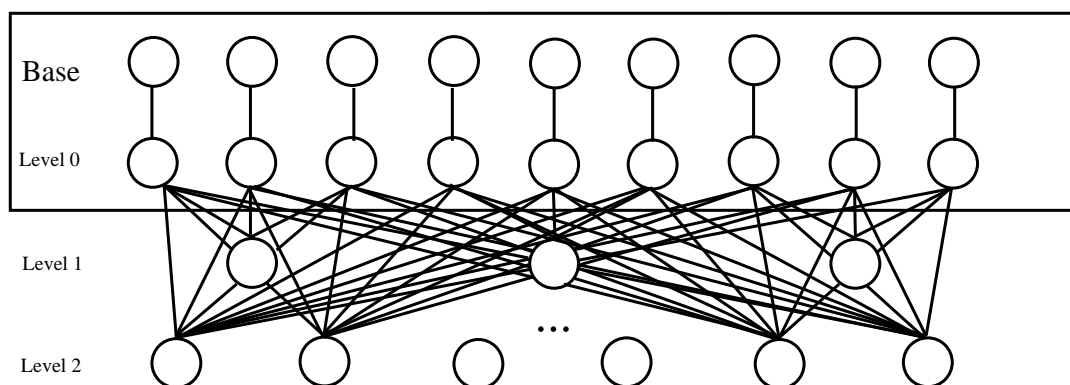

Figure 2: Example for the graph G_2 of the family in exercise 2

connected to) from each pair. So the OPT is 3^n for G_n.

The algorithm instead will take $\Psi(n)$ for G_n. This is, because we constructed the graph the way, that the vertices with the most uncovered edges are the one in the highest level. The base level will have at most one degree less, then the ones in the top level. After alle the toplevel vertices are taken for the Vertex Cover, the resulting graph will be three times G_{n-1}, because we could erase all taken nodes and edges connected to them, because the algorithm will no longer concern them as they all are covered.
The vertices we need for the vertex cover therefore will only be taken, after we took alle extra vertices from every level.

To proof this is not a constant-factor approximation we have to show, that $\frac{\Psi(n)}{3^n}$ is unbounded.

$$
\begin{aligned}
\frac{\Psi(n)}{3^n} &= 3^n + \sum_{k=1}^{n-1} 3^{k-1} \times \eta(n-k-1) \cdot \frac{1}{3^n} \\
&= 1 + \sum_{k=1}^{n-1} 3^{k-1-n} \times \eta(n-k-1)
\end{aligned}
$$

For the result we must only concern ourselves with the sum. As we now through simple analysis these row diverge if the inner sequence does not converge against zero. Because we only sum up positive values the row would diverge against positive infinity.

Claim 4 $\forall n > 0 : \eta(n) = 3^n - 3^{n-1}$

Proof Induction over n

    I.A. $n = 1$
        $\eta(1) = 3^1 - 1 = 3^1 - 3^0$
    I.S. $n \to n+1$

$$
\begin{aligned}
\eta(n+1) &= 3^{n+1} - \left(\sum_{k=0}^{n} \eta(k) \right) - 1 \\
&\overset{\text{I.P.}}{=} 3^{n+1} - \left(\sum_{k=0}^{n} 3^k - 3^{k-1} \right) - 1 \\
&= 3^{n+1} - 3^n - \left(\sum_{k=1}^{n-1} 3^k - 3^k \right) \\
&= 3^{n+1} - 3^n
\end{aligned}
$$

$\square$

If we take this claim into consideration

$$
\begin{aligned}
\frac{\Psi(n)}{3^n} &= 1 + \sum_{k=1}^{n-1} 3^{k-1-n} \cdot ((3^{n-k-1} - 3^{n-k-2}) \\
&= 1 + \sum_{k=1}^{n-1} 3^{k-1-n+n-k-1} - 3^{k-1-n+n-k-2} \\
&= 1 + \sum_{k=1}^{n-1} 3^{-2} - 3^{-3} \\
&= 1 + \sum_{k=1}^{n-1} \frac{2}{27}
\end{aligned}
$$

holds.

The inner sequence is a constant sequence and is never zero. Therefore the row does in fact diverge against positive infinity.

We proofed, that the approximation is not constant. $\qquad\square$

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 5 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.

$\qquad\square$

Claim 6 Let $G(V, E)$ be a graph and $T(V, E')$ be a spanning tree of G.
If $I = \{v \in V \mid \deg_T(v) \geq 2\}$ is a Vertex Cover, then $|I| \leq 2 \times OPT$ holds.

Proof

Let $n = |V|$ the amount of vertices, $i = |I| = |\{v \in V \mid \deg_T(v) \geq 2\}|$ the amount of inner vertices.

Because every vertex except for the root in the tree has exactly one parent. So the amount of tree edges is $n - 1$ and the amount of edges in between the nonleaf vertices is $i - 1$.

Because of that fact the Vertex Cover has to be at least $\frac{i-1}{2}$ large as the amount of inner vertices i. So the optimal solution $OPT \geq \frac{i-1}{2}$. Considering that

$$i \leq 2 \cdot \frac{i - 1}{2} + \frac{1}{2} \leq 2 \cdot OPT + \frac{1}{2}$$

holds, but both the optimal and the feasible solution of $\mathfrak{D}$ are natural Numbers, so the equation

$$i \leq 2 \cdot OPT$$

holds. $\qquad\square$

Claim 7 $\mathfrak{D}$ is a 2-approximation-algorithm.

Proof

In claim 5 we had shown, that the DFS tree's inner vertices are a Vertex Cover. The DFS tree is a Spanning Tree, because it connects every vertex and is itself a tree. Together with claim 6 we now conclude, that the output C of $\mathfrak{D}$ must assert

$$|C| \leq 2 \cdot OPT.$$

$\qquad\square$