**DISCLAIMER:** These notes are not necessarily an accurate representation of what I said during the class. They are mostly what I intend to say, and have not been carefully edited.

In the few weeks before the break, we formalized the notion of NP, NP-hardness, NP-completeness, and discussed approximation algorithms. The notion of approximation algorithms is usually to deal with problems that are hard to solve exactly, specifically NP-hard to solve exactly. In this class, we will discuss such an algorithm for maximum cut, and also justify it by showing that maximum cut is NP-complete.

## 1 2-Approximation for Max-Cut

Recall that a cut in a graph $G$ is the set of edges leaving some set of vertices:

$$E(S, V \setminus S) \overset{\text{def}}{=} \{uv : u \in S, v \notin S\}.$$  

The **MaxCut** problem asks to find a cut with the maximum number of edges.

We start by giving a 2-approximation to **MaxCut**. Recall for a maximization problem, the approximation ratio of algorithm $A$ is dened to be

$$\max_I \frac{\text{OPT}(I)}{A(I)}.$$  

Here a simple greedy algorithm works: label the vertices

$$v_1 \ldots v_n.$$  

For each vertex $v_i$, put $v_i$ on a side different than the majority of its neighbors among $v_1 \ldots v_{i-1}$.

This ensures that we get at least half of $v_i$’s neighbors to smaller labeled vertices. Since each edge $uv$ has a larger labelled end point, we get that we get at least half the edges, aka

$$A(I) \geq m(I)$$  

where $m(I)$ is the number of edges in $I$.

On the other hand, we can at most cut all the edges, so we get

$$\text{OPT}(I) \leq m(I).$$  

So puttingi these together gives for any input $I$

$$\frac{\text{OPT}(I)}{A(I)} \leq \frac{m}{m/2} \leq 2,$$

so this is a 2-approximation.
2 VertexCover $\rightarrow$ MaxCut

We now show that MaxCut is NP-hard by reducing VertexCover to it. Combining this with checking that it’s in NP gives that MaxCut is also NP-complete.

The construction is as follows: given a graph $G$ with vertices $v_1 \ldots v_n$, with degrees $deg(v_1) \ldots deg(v_n)$, we create a new graph $H$ by

1. adding a new vertex $w$, and
2. connect $w$ with each $v_i$ by $deg(v_i) - 1$ parallel edges.

We claim $H$ has a cut of size $2m - k$ if and only if $G$ has an independent set of size $k$.

Consider a cut in $H$. Let $S$ be the half that does not contain $w$. The number of edges cut is

$$\sum_{v \in S} (deg(v) - 1) + |uv : u \in S, v \notin S| = \sum_{v \in S} deg(v) + |uv : u \in S, v \notin S| - |S|.$$ 

Each edge $uv$ with both end points in $S$ gets counted twice: once per degree term. Each edge $uv$ with one end point in $S$ also gets counted twice: once in the degree term, and once among the number of edges cut. So the above expression equals to

$$2 \cdot |\text{number of edges incident to } S| - |S|.$$ 

Note that if we have an edge with both endpoints outside of $S$, we can always increase the above objective by adding one of its endpoints to $S$. Therefore the only $S$ we need to consider are independent sets. And for those, the objective becomes

$$2m - |S|,$$

which is at least $2m - k$ if and only if $|S| \leq k$. 