MATH 8500 Algorithmic Graph Theory, Spring 2017, OSU
Lecture 3: Max-Cut and Szemeredi's Regularity Lemma
Instructor: Anastasios Sidiropoulos
Scribe: Dhanvi Athmakuri

1 Problem Definition

The MAX-CUT Problem is similar to the MIN-CUT Problem, except that here the goal is to maximise rather than to minimize the size of the cut. Similar to the earlier case, we consider an un-weighted graph. So, the goal of the problem translates to finding the partition of the vertices into sets, by cutting through the maximum number of edges.

The formal definition of the problem is as follows :
Definition :
l-way cut / Max-cut :
input : $G = (V, E)$ (unweighted and connected) and $|V| = n$
Goal : Find a partition $S = S_1, S_2, ...S_l$ of V, which maximizes $|E(S)|$, where

$$E(S) = \{\{u, v\} \in E : u \in S_i \ \& \ v \in S_j, \text{ for some } i \neq j\}$$

The following is an abstract pictorial depiction of the max-cut problem.

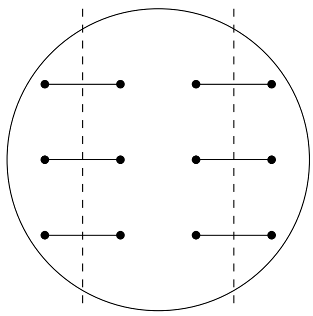


Figure 1: abstract depiction of l-way cut (l=3)
The circle represents an arbitrary graph

2 Properties and Results

This problem is NP-hard even for $l = 2$. The following re-states the same.

Theorem : The two-way cut problem (Max-cut Problem) is NP-hard.

For an illustration of the solution to this problem, consider the following graph:

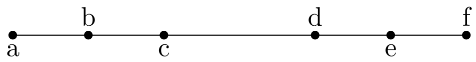


Figure 2: Example graph

Considering all the edges to be unweighted, the solution to the MAX-CUT problem (l=2) for the graph in fig 2, is $S_1 = \{a, c, e\}$ and $S_2 = \{b, d, f\}$. That is, we cut through as many edges as possible, to partition the vertex set.

Since the MAX-CUT problem is NP-hard, we attempt to design an **approximation algorithm** for the problem.

claim : There always exists a partition S such that,

$$|E(S)| \geq (1 - \frac{1}{l})|E|$$

Essentially, we are claiming that there exists an approximation algorithm, which produces a *solution* S, such that $|E(S)| \geq (1 - \frac{1}{l})|E|$

The above claim can be proved constructively, by considering the following algorithm :

ALGORITHM : APPROX

1. **input :** $G(V, E)$ (connected and un-weighted)
2. **output :** set S of edges approximating the maximal cut for G.
3. for each $v \in V$, pick $i(v) \in \{1, 2, 3....l\}$, uniformly at random & independently and place v in $S_{i(v)}$
4. **return** $E(S) = \{\{u, v\} \in E : u \in S_i \ \& \ v \in S_j, for\ some\ i \neq j\}$

We will now see that this algorithm produces a solution that satisfies the claim made earlier.

proof : $\forall\ u \neq v \in V$, we have

$$P\{i(u) = i(v)\} = \frac{1}{l}$$

$$\forall\ (u, v) \in E,$$

$$Let\ X_{(u,v)} = \begin{cases} 1 & if\ (u, v)\ is\ cut \\ 0 & otherwise \end{cases}$$

It is easy to see that

$$|E(S)| = \sum_{(u,v) \in E} X_{(u,v)}$$

This is because, $X_{(u,v)}$ is 0, when (u,v) is cut and 1, otherwise. Therefore, $\sum_{(u,v)\ \in\ E} X_{(u,v)}$, gives the number of edges which are included in the set S.
Here, we have to understand that $|E(S)|$ is a random variable because $X_{(u,v)}$ is a random variable. Therefore, we take *Expectation* on both sides. We then get

$$E[|E(S)|] \ = \ E[\sum_{(u,v)\ \in\ E} X_{(u,v)}]$$

By the property of *Linearity of Expectations*, we get,

$$= \sum_{(u,v)\ \in\ E} E[X_{(u,v)}]$$

$$= \sum_{(u,v)\ \in\ E} (1 - \frac{1}{l})$$

This is because, $E[X_{(u,v)}] \ = \ 1 \times \ P\{X_{(u,v)} = 1\} \ + \ 0 \ \times \ P\{X(u,v) = 0\} \ = \ P\{X_{(u,v)} = 1\} \ = \ 1 - P\{X_{(u,v)\}=0} \ = \ (1 - \frac{1}{l})$

We therefore have,

$$E[|E(S)|] \ = \ (1 - \frac{1}{l})|E|$$

Since the above value is an expectation, we will have partitions that produce values greater than the above value and also partitions that produce values lesser than the above value. Thus, we can safely say that there exists *some* partition that produces value at least as much as the above value.

Thus, we have proved our claim that there exists an algorithm which produces a solution such that $|E(S)| \ \geq \ (1 - \frac{1}{l})|E|$

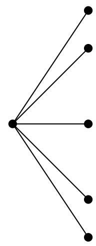
Shortly, we will define a very important lemma, and get an intuitive understanding of its meaning and implications. But before that we consider two cases, in which the algorithm APPROX, fails by a significant factor (We will consider examples only for $l = 2$). Consider the following graphs given by figure 3a and 3b.

Consider the complete Bipartite graph in figure 3b. Here, the maximum cut is obtained by cutting through all edges. Where as, the expected number of edges obtained by the algorithm APPROX is $\frac{|E|}{2}$, where $|E|$ is the number of edges. The algorithm is off by a factor of 2. The case of figure 3a caan be understood with the similar intuition. The reader is requested to take some time and digest this fact.

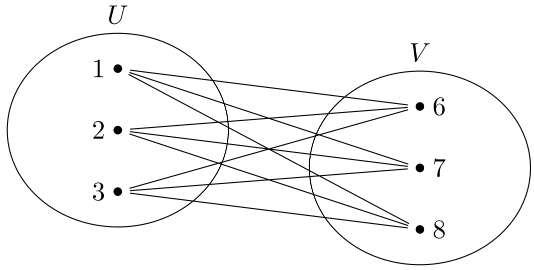
Next, we will consider a Lemma of great importance in combinatorial math and extremal graph theory, called the **Szemeredi's Regularity Lemma**.

Szemeredi's Regularity Lemma :
$\forall \epsilon > 0, \forall m \in \mathbb{N}, \exists P(\epsilon, m)$ & $Q(\epsilon, m) \in \mathbb{N}$, such that $\forall G(V,E)$, with $n \geq P(\epsilon, m), \exists$ a partition, $V_1, V_2, ...V_k$ of V such that

1. $m \leq k \leq Q(\epsilon, m)$
2. $\lceil \frac{n}{k} \rceil - 1 \leq |V_i| \leq \lceil \frac{n}{k} \rceil$
3. All but ϵk^2 of the pairs (V_i, V_i) are $\epsilon - regular$



(a)



(b) complete bipartite graph

Figure 3

The classes V_i will be called **groups** or **clusters**. The notion of *ϵregularity* is formalized as follows :

Definition : Let $G(V, E)$ and $A, B \subseteq V$ and $A \cap B = \phi$. Let

$$e(A, B) = |E(A, B)| \ (number\ of\ edges\ between\ A\&\ B)$$

Then, we define

$$d(A, B) = \frac{e(A, B)}{|A||B|}$$

where, $d(A, B)$ is the density of edges between A & B.

We say that A & B are $\epsilon - regular$, if :

$$\forall X \subseteq A,\ with\ |X| \geq \epsilon|A|$$
$$\forall Y \subseteq B,\ with\ |Y| \geq \epsilon|B|$$

we have

$$|d(X, Y) - d(A, B)| < \epsilon$$

Think of the ϵ-regular pairs as highly uniform bipartite graphs. Namely, the ones in which the density of any *reasonably* sized sub-graph is about the same as the overall density of the graph.

The Regularity Lemma says that every dense graph can be partitioned into a small number of regular pairs and a few left-over edges. Since regular pairs behave as random bipartite graphs in

many ways, The Regularity lemma provides us with an approximation of a large dense graph, with the union of a small number of random looking bipartite graphs.
If we delete the edges within clusters as well as edges that belong to irregular pairs (pairs which are not ϵ-regular), of the partition, we get a subgraph $G^{'} \subseteq G$, that is more uniform, more random-looking and therefore more manageable. Since the number of edges deleted is small compared to $|V|^2$, the Regularity Lemma provides us with a good approximation of G, by the random-looking graph $G^{'}$