CSL861: Algorithmic Graph Theory **Semester I 2013**

Lecture 10: November 20

Lecturer: Prof. Amit Kumar *Scribes: Davis Issac*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

In the last lecture we saw that:

- A graph $G(V, E)$ is caled an (A, d) *expander* if $N(S) \geq d|S|$ for all $S \subseteq V$ such that $|S| \leq A$.
- Given any δ, there exist an $(\frac{n}{\delta}, \delta - 2)$ expander where every vertex has degree δ. We had proved for a weaker value than $\frac{n}{\delta}$ but it can be proved for $\frac{n}{\delta}$. Also, we had given a randomized construction but there are deterministic constructions known.

In this lecture we will see two applications of the expanders, one in *derandomization* and other in *disjoint path routing.*

In most applications, we use expanders with exponential number of verices. This is feasible because we will not be explicitly storing the whole expander graph. Each vertex can be represented using $O(n)$ space. Each vertex can be thought of as an n-bit number. Many constructions give a way to quickly find the neighbours of a vertex, given the vertex.

10.1 Applications of Expanders

10.1.1 In Derandomization

By derandomization, we mean reducing the error of a randomized algorithm.

Suppose, we have a randomized algorithm **A** which tests the membership in some language L. **A** takes an input x and outputs YES or NO. On an input x, A behaves as:

- if $x \in L$, **A** outputs YES with a probablity of $\frac{1}{2}$.
- if $x \notin L$, **A** always outputs NO.

Suppose, **A** uses r random bits where $r = poly(|x|)$.

Our objective is to reduce the error probablity from $\frac{1}{2}$ to δ.

One way to do it is the following procedure:

- Repeat **A** for k times.

- If **A** says YES atleast once then output YES, otherwise output NO.

The above algorithm gives error output only if **A** gives error output on each of the k times. So, the total error probablity is $(\frac{1}{2})^k$. So, inorder to get an error probablity of less than δ, we should take $k = \log(\frac{1}{\delta})$. So, the total number of random bits needed is $r \log(\frac{1}{\delta})$.

Can we achieve the same error probablity without using so many random bits? By using expanders, this can be done using r random bits as shown below.

Let G be a $(\frac{n}{4}, 2)$ expander on 2^r vertices. Each vertex in G corresponds to a sequence of r bits. The algorithm is as follows:

- Pick a vertex v of G uniformly at random.(This uses r random bits).
- Let $X = \{w \in G \mid w \text{ is at a distance} \leq c \text{ from } v\}$ where c is a constant. (Note that $|X| \leq 4^c$ in this case)
- Run **A** on each $v \in X$.

We will prove that the error probablity of this algorithm is very low. Let $N = 2^r$ denote the total number of vertices. Let Y be the set of vertices which gives a wrong answer for **A**. $|Y| \leq \frac{N}{2}$ because **A** has an error probablity of atmost $\frac{1}{2}$. Our algorithm gives a wrong answer only if all the vertices in X are in Y. For this to happen all vertices within a distance of c from the randomly selected vertex v should be in Y. But, since G is an expander, the chance of all these vertices being confined to Y is very low.

Let us prove this formally. Let B be the set of vertices $y \in Y$ such that $N(y) \subseteq Y$. If $|B| > \frac{N}{4}$, then $|N(B)| > \frac{N}{2}$ as G is an $(\frac{N}{4}, 2)$ expander. So, if $|B| > \frac{N}{4}$, B will have a neighbour outside Y. Hence, B can have only size of atmost $\frac{N}{4}$. Similiarly ,if C is the set of vertices in B having all its neighbours inside B ,then $|C| \leq \frac{N}{8}$. In general, any set of vertices in Y such that all the vertices at a distance $\leq c$ from it are in Y can have a size of atmost $\frac{N}{2^c}$. Hence, we get that the error probablity is atmost $\frac{1}{2^c}$. Inorder to get an error of less than δ, we should take $c = \log(\frac{1}{\delta})$. So, algorithm **A** is repeated a total of $4^{\log \frac{1}{\delta}} = \frac{1}{\delta^2}$ times.

Hence, although the running time increases, we are able to achieve the required error probablity with only r random bits.

10.1.2 Disjoint Path Routing

We have n vertices on both sides of a routing network. Let X be the vertices on the left and Y be the vertices on right. We have to connect them using a suitable graph such that for any $T \subseteq X$ and $S \subseteq Y$ with $|T| = |S| = k$, there are k disjoint paths between T and S(see figure 10.1).

Note that this can be easiliy done using a complete bipartite graph on X and Y. But we are interested in finding such a graph with $O(n)$ edges. Such a graph is also called a **super concentrator**.

We will use a particular type of expander for building the super concentrator. The exapander that we will use is a bipartite graph G on vertex sets L and R(see figure 10.2). L contains n vertices and R contains $\frac{3n}{4}$

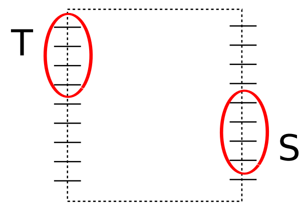

Figure 10.1: Disjoint Path Routing

vertices. G satisfies the following property : any $S \subseteq L$ such that $|S| \leq \frac{n}{2}$ has $|N(S)| \geq |S|$.
Exercise : Check that if every vertex in L selects 10 random neighbours from R uniformly, the above property holds with high probablity.

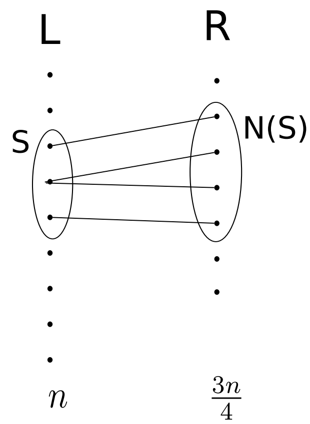

Figure 10.2: Expander used to build super concentrator

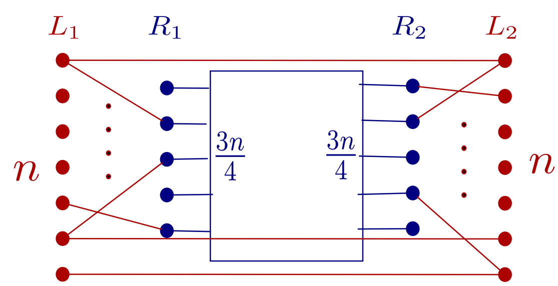

Figure 10.3: building (n, n) super concentrator using a $(\frac{3n}{4}, \frac{3n}{4})$ super concentrator

We will build the super concentrator recursively. First, we will assume that a $(\frac{3n}{4}, \frac{3n}{4})$ super concentrator is available and then build an (n, n) super concentrator using this. Then, the $(\frac{3n}{4}, \frac{3n}{4})$ super concentrator can be built recursively in the same manner.

Let R_1 and R_2 be the vertices on the two sides of the $(\frac{3n}{4}, \frac{3n}{4})$ super concentrator. Let L_1 and L_2 be the two sets of n vertices that need to be connected using (n, n) super concentrator. Connect L_1 with R_1 such that the expander property discussed above is satisfied. Connect L_2 and R_2 in the same manner. Also, add

n direct edges from L_1 to L_2 such that i^{th} vertex in L_1 is connected to i^{th} vertex in L_2(see figure 10.3).

The total number of edges,$E(n) = 10n + 10n + n + E(\frac{3n}{4})$. Hence, $E(n) = O(n)$.

Let $T \subseteq L_1$ and $S \subseteq L_2$ such that $|T| = |S| = k$. We will show that there exist k disjoint paths between T and S. Suppose $k \leq \frac{n}{2}$. By the property of the expander, for any $T' \subseteq T$, $|N(T')| \geq |T'|$. Here, $N(T')$ denotes the neighbours of T' in R_1 . Hence, by *Hall's theorem*, T has a matching in R_1. Let T_1 denote these matched vertices in R_1. Similiarly S has a matching in R_2. Let S_2 denote these matched vertices in R_2. But by the property of the $(\frac{3n}{4}, \frac{3n}{4})$ super concentrator, there exist k disjoint paths between T_1 and S_2. By adding both matchings with these paths we get k disjoint paths between S and T. If $k > \frac{n}{2}$, then atleast $k - \frac{n}{2}$ pairs of vertices can be connected through the direct edges between L_1 and L_2. The remaining $\frac{n}{2}$ pairs can be connected in the same way as in the case when $k \leq \frac{n}{2}$.