CS 683: Advanced Algorithms **Applications of Approximate Sparsest Cut**
Thu, 02/08/01 **Scribe: Mikhail Kobyakov, Martin Pal**

1 Introduction

Algorithms for finding approximate sparsest cuts can be used to solve other graph partitioning problems arising for example in VLSI circuit design. The idea is often to divide the graph into two balanced pieces and then employ the divide and conquer paradigm.

In this lecture, we will be interested in the following problem: Given a graph G, $|G| = n$, we want to map its vertices onto the set $\{1, 2, \ldots, n\}$ of integers. We may think of it as designing a bus with n slots, each vertex corresponds to a device, edges correspond to pairs of devices that need to communicate. We want assign devices to different slots on the bus so that devices that heavily communicate are "close" to each other. Formally, a *linear layout* of a graph G, $|G| = n$ is a bijection $l : V(G) \to \{1, \ldots, n\}$. Throughout the lecture, the size of the graph G will be denoted by n. We concentrate on two measures:

- the *cutwidth* of a linear layout l of a graph G is the size of a maximum size cut separating U and $V \setminus U$, where $U = \{v | l(v) \leq i\}$ for some i. It is the maximum number of wires passing through a single point of the bus in parallel.
- the *wire length* of a layout is $\sum_{(u,v)\in E(G)} |l(u) - l(v)|$, the total length of wires between all pairs of nodes.

For example, the layout on figure 1 has cutwidth 2 and wirelength 9.

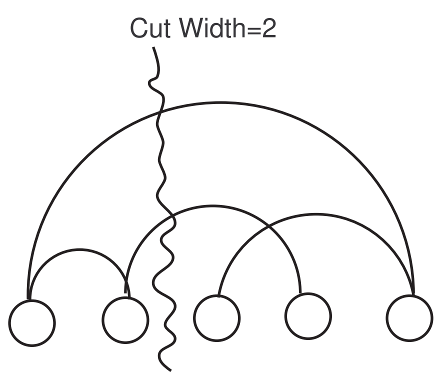

Figure 1: a bus - n dots on a line

Before trying to solve the mentioned problems, we take a look at approximating graph bisection. We build on the results from last lecture, where we have seen an $O(\log n)$ approximation algorithm for min-ratio-cut (minimize $\frac{e(U,V\setminus U)}{|U|\cdot|V\setminus U|}$ over all subsets $U \subseteq V$).

2 Approximating Graph Bisection

The problem is as follows: Given graph G, find a set U, $|U| = \lfloor \frac{n}{2} \rfloor$ that minimizes $e(U, V \setminus U)$, the number of edges between U and $V \setminus U$. This problem is known to be NP hard. A rough idea for an approximation algorithm would be: Use the approximation algorithm from previous lecture for min-ratio cut. If it splits the graph into two roughly equal parts, we are happy. If not, apply the algorithm to the bigger part. Keep chopping slices off the biggest chunk until the remaining part is roughly half of the original graph. Put all the small pieces into one partition and the big chunk into the second.

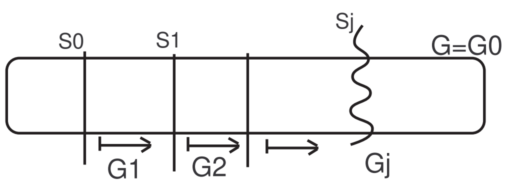

Figure 2: a graph with pieces sliced off

The problem with this approach is that we don't get the graph spliced into exact halves. To get around, we relax our requirements slightly. A *b-balanced separator* is a set $U \subseteq V$ such that $|U|, |V \setminus U| \geq bn$. We proceed to show that an algorithm based on the idea above finds a $\frac{1}{3}$-balanced separator whose cut size is within a factor of $O(\log n)$ of the size of minimal bisection.

Our algorithm works as follows. We construct a sequence of graphs $G = G_0 \supseteq G_1 \supseteq \ldots \supseteq G_k$, such that $\frac{n}{3} \leq |G_k| \leq \frac{2n}{3}$. For each $i \geq 0$, we find an approximate min-ratio cut in the graph G_i, let G_{i+1} be the bigger partition of G_i and recur. We stop when the size of some G_i drops below $\frac{2n}{3}$ and call this graph G_k. Clearly, $|G_k| \geq \frac{n}{3}$. Let s_i $(0 \leq i < k)$ be the number of edges cut while cutting graph G_i and let $\alpha_i n$ be the size of $G_i \setminus G_{i+1}$.

Let S^* be the number od edges in the minimum bisection cut. Since the size of each G_i, $i < k$ is at least $\frac{2}{3}n$, it must contain at least $\frac{n}{6}$ nodes from each partition of the bisection of G. In other words, if we take the set of edges from the minimum bisection of G, it induces a cut in each G_i with cut ratio at most $S^* \frac{6}{n} \frac{2}{n} = s^* \frac{12}{n^2}$. This is clearly an upper bound on the ratio of the min-ratio cut.

The cut ratios of our cut i is $\frac{s_i}{\alpha_i n(|G_i| - \alpha_i n)}$, which is bounded from below by $\frac{s_i}{\alpha_i n^2}$. Since this cut is within $O(\log n)$ factor of the optimal min-ratio cut,

$$\frac{s_i}{\alpha_i n^2} = O(S^* \frac{12}{n^2})$$

Solving for s_i and summing over all i we obtain

$$\sum_{i=0}^{k-1} s_i \leq \sum_{i=0}^{k-1} \alpha_i (12cS^* \log n + d)$$

where c and d are some constants form the big-oh notation. Since $\sum_{i=0}^{k-1} \alpha_i \leq 1$, we get (hiding constants again) $\sum_{i=0}^{k-1} s_i = O(S^* \log n)$.

We can view bisection of a graph as a $\frac{1}{2}$-balanced separator. Our algorithm finds a $\frac{1}{3}$-separator that is within $O(\log n)$ factor of the minimum $\frac{1}{2}$ separator. We can slightly generalize it to solve the following problem: Let S_b^* be the minimum number of edges in a b-balanced separator of graph G. For a $b' < b$, $b' \leq \frac{1}{3}$, find a b' balanced separator with number of edges within a factor of $O(\log n)$ of S_b^*. The only modification we need to make to our algorithm is to stop splitting after the size of G_i drops below b' instead of $\frac{1}{3}$. Using basically the same analysis as above, we can show that the size of the cut found by the modified algorithm is

$$O(\frac{S_b^* \log n}{b' - b})$$

3 Linear Layout - The Algorithm

We now present a simple and natural approximation algorithm for linear layout. We use an divide and conquer approach. We find a b' (say, $b' = \frac{1}{4}$) balanced separator of G using the algorithm from previous section. We then recur to both subgraphs, finding an approximately optimal layout for both subgraphs, and then glue together the layouts of the "left" and "right" partition. We choose arbitrarily which partition is left and right respectively. Despite the simplicity of this algorithm we can prove surprising approximation guarantees for both cutwidth and wirelength objectives. Observe that since b' is a constant, the depth of the recursion will be $O(\log n)$.

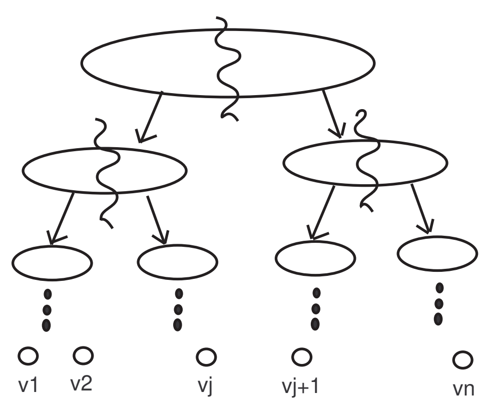

Figure 3: a tree of graphs

4 Cutwidth

Let $C^*(G)$ be the minimum cutwidth of a graph G and let $S^*(G)$ be the size of its minimal bisection. Clearly $C^*(G) \geq S^*(G)$. Even more, $C^*(G) \geq \max_{G' \subseteq G} S^*(G)$. Let $S_{\max} = \max_{G' \subseteq G} S^*(G)$. In each recursive call of our algorithm we make a cut of size $O(S_{\max} \log n)$. Now consider a point on the bus between slots i and $i+1$ for some i. Consider all the graphs on which the algorithm was recursively invoked during the whole execution. What we mean is this. Let $X(G)$ be a set of subgraphs of G defined inductively. If G is split into G_l and G_r, then $X(G) = \{G\} \cup X(G_l) \cup X(G_r)$, if G is the trivial one vertex graph let $X(G) = \{G\}$. Now let u and v be the vertices of G assigned to slots i and $i+1$. Let X' be the set of all graphs $H \in X$ that contain vertices u and v. These graphs form a chain $H_1 \supseteq H_2 \supseteq \ldots \supseteq H_k$ for some $k = O(\log n)$. Realize that every edge passing between slots i and $i+1$ must have been cut when splitting one of the graphs from X'. When splitting H_j, at most $O(S_{\max} \log n)$ edges were cut. There are $O(\log n)$ graphs in X', which gives an upper bound on the total number of edges cut $O(S_{\max} \log^2 n)$. Thus we have

Lemma 4.1 *The algorithm above finds a linear layout with cutwidth within a factor of $O(\log^2 n)$ from optimal.*

5 Wirelength

For wirelength, the case of a bus of expander graphs presents a problem, as it is conceivable that some expander graph will be cut with an aproximate cut, and the wires will be stretched out. However, it will turn out that the problem is avoidable.

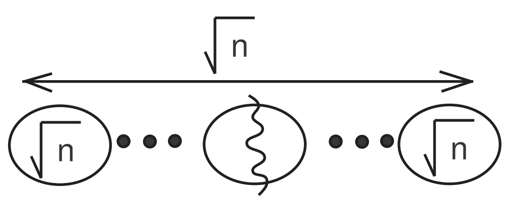

Figure 4: a bus of expander graphs

For the sake of concreteness, assume $b = \frac{1}{3}$. Take some constant $b' > b$ and use our divide and conquer algorithm to design a linear layout for a graph G. We claim that

Lemma 5.1 *The algorithm finds a linear layout with wirelength within a factor of $O(\log^2 n)$ from optimal.*

Before attempting to prove the lemma, we make two easy observtions. Let $w^*(G)$ denote the minimal wirelength of a graph G. Let $S^*_{\frac{1}{3}}(G)$ be the size of a minimal $\frac{1}{3}$ balanced separator of G.

Claim 5.2 *Let $H_1, \ldots, H_k$ be disjoint subgraphs of G. Then $w^*(G) \geq \sum_{i=1}^{k} w^*(H_i)$.*

Proof. Every layout of G gives a layout of the graph $\bigcup_i H_i$ with wirelength not exceeding the wirelength of G, since $\bigcup_i H_i$ is a subgraph of G. ∎

Claim 5.3 *For any G, $w^*(G) \geq \frac{1}{3} n S^*_{\frac{1}{3}}(G)$*

Proof. Take a linear layout of G. Consider the $\frac{n}{3}$ cuts between slots $\frac{n}{3}$ and $\frac{n}{3}+1$, $\frac{n}{3}+1$ and $\frac{n}{3}+2$ up to $\frac{2n}{3}$ and $\frac{2n}{3}+1$ (for simplicity assume n is a multiple of 3). Any of these cuts has size at least $S^*_{\frac{1}{3}}(G)$, the presence of an edge passing through each interval adds 1 to the total wirelength and there are $\frac{n}{3}$ intervals. ∎

Proof of lemma 5.1. Consider the tree of graphs from $X(G)$ (defined in previous section, see fig. 3). Consider all H_i on a particular level l of this tree. Each of these graphs is cut into two subgraphs belonging to level $l+1$. By claim 5.3, the size of the cut of each H_i times the size of H_i is within $O(\log n)$ factor from the optimal wirelength of the graph H_i (because the cut of H_i is within $O(\log n)$ from $S^*_{\frac{1}{3}}(H_i)$). Consider an edge cut while cutting a graph H_i. Its length in the resulting layout and thus its contribution to the total wirelength is at most $|H_i|$. Counting all the contributions in one level of the tree

$$
\begin{aligned}
\sum_{i=1}^{2^l} \text{contribution of edges of } H_i \text{ cut on level } l \quad &\leq \\
\sum_{i=1}^{2^l} |H_i| \cdot \text{number of edges cut in } H_i \text{ at level } l \quad &\leq \\
\sum_{i=1}^{2^l} w^*(H_i) \log |H_i| \quad &\leq \quad w^*(G) \log n
\end{aligned}
$$

Realize that every edge is cut at least (exactly) once, and thus contributes to the sum at some level. There are $O(\log n)$ levels, so summing up the above inequalities over all levels gives the desired result

$$\text{contribution of all edges of } G \leq w^*(G) \log^2 n.$$

∎