CPSC 665 : An Algorithmist's Toolkit Lecture 17 : 23 Mar 2015

Low-Stretch Spanning Trees

Lecturer: Sushant Sachdeva *Scribe: Alex Reinking*

1. Introduction

In this lecture we discussed the results of one of Professor Spielman's papers. The Star Decomposition algorithm takes a graph and recursively divides it into smaller parts until we create a low-stretch spanning tree. Most of the algorithm is easiest to understand by looking at an illustration, so much of these notes are devoted to presenting diagrams, rather than technical details.

2. Preliminaries

Definition 2.1 (Stretch)**.** *Given a graph G and a spanning tree $T \subseteq G = (V, E)$, define the stretch of T as*

$$\mathsf{Stretch}(T) = \sum_{e \in G \setminus T} |c_e| \tag{1}$$

where c_e is the unique cycle in $T \cup \{e\}$. That is, for each edge in the graph, we look at the new shortest path in the tree corresponding to the edge (think of the edge as being "stretched" onto the tree). So if we let $d_G(u, v)$ be the distance between $u, v \in V$ in G, and $d_S(u, v)$ be the distance between $u, v \in S \subseteq V$ in the subgraph induced by S, then, in fact:

$$\mathsf{Stretch}(T) = \sum_{e=(u,v) \in G \setminus T} 1 + d_T(u, v) \tag{2}$$

So, we would like to find a tree T of low total stretch, ideally on the order of $O(m\ \mathsf{poly}(\log n))$ where m is the number of edges in G and n is the number of vertices. The Star Decomposition algorithm, devised by Elkin, Emek, Spielman, and Teng [EEST08], produces a tree of stretch $O(m \log^3 n)$ in time $O(m \log m)$.

3. Example

Star graphs, shown in figure 1, are a sort of best-case scenario for stretch. Like all trees, their total stretch is 0, but they also have the nice property that the longest distance between two vertices in the graph is 2 (for S_3 and up). Running the star decomposition algorithm on a star graph predictably yields the same graph in response.

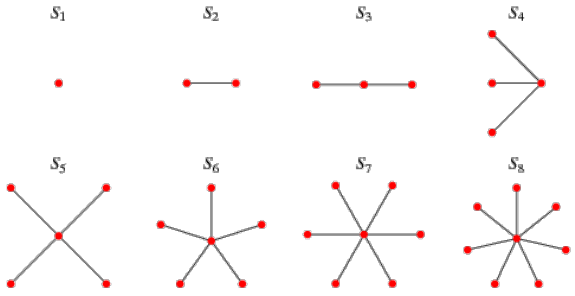

Figure 1. Star graphs of a few orders. (Source: MathWorld)

4. Star Decomposition Properties

In this section, we will present the Star Decomposition algorithm through a series of loose pictures and proofs, with the aim of getting the intuition for the algorithm across, rather than the technical details.

4.1. **Inputs and Outputs.** The Star Decomposition algorithm takes in a graph G and a distinguished vertex x_0. It produces a partition of V, $(V_0, V_1, \ldots, V_k)$, so that

(1) The V_i are a partition ie. disjoint and $V = \bigcup_{i=0}^{k} V_i$
(2) $x_0 \in V_0$
(3) The V_i are connected so that $\exists (x_i, y_i) \forall i = 1, \ldots, k$ such that $x_i \in V_i$ and $y_i \in V_0$

This situation is depicted in figure 2.

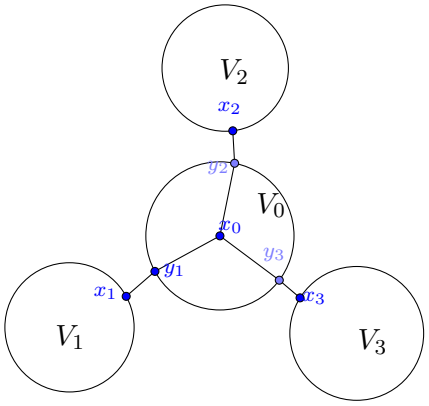

Figure 2. Illustration of Star Decomposition output

4.2. **Properties.** Now, we'll move on to defining a few desirable properties for our output to have. We will use these properties to show that the stretch is low. First, recall the definition for the radius around a vertex in a graph:

Definition 4.1 (Radius in a Graph)**.** *The radius is rooted on a vertex x and is defined to be the distance to the furthest vertex in the graph. Formally:* $\mathrm{rad}_{V_i}(x) = \max_{u \in V_i} d_{V_i}(x, u)$

The first additional property we would like to have is balance. We say a partition is **balanced** when all of the parts are some constant fraction smaller than the whole. We take 3/4 to be a fraction suitable for our purposes. Formally:

$$\mathrm{rad}_{V_i}(x_i) \leq \frac{3}{4} \mathrm{rad}_V(x_0) \tag{3}$$

Secondly, we would like the partition to be ϵ-efficient, by which we mean that if $G' = G - \{\text{edges between } V_i \text{ and } V_j \forall i \neq j\} + \{(x_i, y_i)\}$ then for all u, $d_{G'}(u, x_0) \leq (1 + \epsilon) d_G(u, x_0)$. This is the condition that will guarantee our low stretch requirement. Basically it's saying that the distance between partitions via the base part V_0 doesn't get stretched too much.

Lemma 4.2. *In $O(m)$ time, we can find an ϵ-efficient, balanced star decomposition such that*

$$|\#\ \textit{of edges cut}| = \left|G \setminus G'\right| \leq O\left(\frac{m}{\epsilon} \frac{\log m}{\mathrm{rad}_G(x_0}\right) \tag{4}$$

Proof. Omitted. See [EEST08] for the full proof. □

5. Producing a Spanning Tree

The method by which we actually produce a spanning tree is straight-forward. We fix a given $x_0 \in G$ and take (V) to be our initial partition. We then recursively apply the star decomposition algorithm to each part of the partition, discarding edges between the resulting parts (as in G'). This gives a sequence of graphs: $G_0 = G, G_1 = G', G_2, \ldots, G_j = T$, which ends in the spanning tree we desire.

5.1. **Claims.** First, that this process ends in a spanning tree is clear. Next, since the radius of each part, which is always between 1 and m, is constantly decreasing by a factor of at most $\frac{3}{4}$, we know there are at most $\frac{\log m}{\log 4/3}$ levels of recursion. Finally, by repeating the ϵ-efficiency condition we get

$$d_{G_1}(u, x_0) \leq (1+\epsilon) d_G(u, x_0)$$
$$d_{G_2}(u, x_0) \leq (1+\epsilon)^2 d_G(u, x_0)$$

and thanks to the radius decrease:

$$d_T(u, x_0) \leq (1+\epsilon)^{\log_{4/3} m} d_G(u, x_0)$$

taking $\epsilon = 1/m$:

$$\leq e \cdot d_G(u, x_0)$$

5.2. **Bounding the stretch.** Now, we will show that the total stretch of the tree is on the order of $O(m \log^3 m)$

Proof. Let H be one of the star pieces from the first decomposition step, say V_2. Now, let $(u, v) \in G'$ be one of the edges thrown away. Then the stretch for this edge is bounded by:

$$\begin{aligned} d_T(u, v) &\leq d_T(u, x_2) + d_T(x_2, v) \\ &\leq e \cdot (d_H(u, x_2) + d_H(x_2, v)) \\ &\leq 2e \cdot \mathrm{rad}_H(x_2) \end{aligned}$$

Then the total stretch of the edges deleted in H during the step from G_2 to G_3 is

$$\sum_{(u,v) \text{ as above}} d_T(u, v) \leq 2e \cdot \mathrm{rad}_H(x_2) \cdot O\left(\frac{m'}{e} \log \frac{m'}{\mathrm{rad}_H x_2}\right) \leq O\left(\frac{m'}{e} \log m'\right) \tag{5}$$

where $m' = \big|E(H)\big|$. Then applied to the whole graph for this phase, the stretch is $O\left(\frac{m}{e} \log m\right)$. Finally, since there are $\log m$ phases at most and $\epsilon = 1/\log m$, we get $O(m \log^3 m)$ for our total stretch. □

6. Producing the Decomposition: A Sketch

Now we have reduced the problem to finding these star decompositions, since if they exist with the properties we prescribed, we will have a low-stretch spanning tree. Our requirements are twofold: we need not only that a low stretch decomposition be produced, it must also be produced efficiently.

6.1. **Region Growing.** We will start to describe the process for region growing by introducing a few definitions. We define a family of concentric sets $\mathcal{L}_r$ for $r \geq 0$ so that $\mathcal{L}_r = B(x_0, r) = \big\{u \parallel d_G(x_0, u) \leq r\big\}$. These sets have the following properties:

(1) $\mathcal{L}_0 = \{\}$
(2) $\mathcal{L}_r \subseteq \mathcal{L}_{r'}$ whenever $r' \geq r$
(3) If $u \in \mathcal{L}_r$ and $(u, v) \in E$, then $v \in \mathcal{L}_{r+d(u,v)}$.

We also say that $\mathrm{vol}(S) = \sum_{u \in S} \deg(u)$, so $\mathrm{vol}(V) = 2m$ and $|\partial S|$ is the number of edges with exactly one endpoint in S.

Lemma 6.1. *Given $0 \leq \lambda < \lambda'$ there exists some $r \in [\lambda, \lambda')$ so that*

$$|\partial \mathcal{L}_r| \leq \frac{1 + \operatorname{vol} \mathcal{L}_r}{\lambda' - \lambda} \log(m+1) \tag{6}$$

This is to say that we can actually find a radius with a small boundary.

Proof. First, we define the **continuous volume** to be $\operatorname{cvol}(r) = 1+$ # of edges inside $\mathcal{L}_r+$ the "fraction" of edges on the boundary within the ball. Take as fact the following two properties of the continuous volume:

(1) $\operatorname{cvol}(r) \leq 1 + \operatorname{vol}(\mathcal{L}_r)$
(2) $\frac{d}{dr} \operatorname{cvol}(r) = |\partial \mathcal{L}_r|$

Then, assume that:

$$\begin{aligned} & |\partial \mathcal{L}_r| > \alpha(1 + \operatorname{cvol}(r)) \\ \Rightarrow & \frac{d}{dr} \operatorname{cvol}(r) > \alpha \operatorname{cvol}(r) \\ \Rightarrow & \int_{\lambda'}^{\lambda} d(\log \operatorname{cvol}(r)) dr > \alpha \int_{\lambda'}^{\lambda} dr = \log \frac{\operatorname{cvol}(\lambda')}{\operatorname{cvol}(\lambda)} \end{aligned}$$

So now $O(\log m) \geq \log \frac{\operatorname{cvol}(\lambda')}{\operatorname{cvol}(\lambda)} > \alpha(\lambda - \lambda')$ is a contradiction for $\alpha = \log(1+2m)/(\lambda - \lambda')$. So, let $\lambda = \mu/3$ and $\lambda' = 2\mu/3$, where $\mu = \operatorname{rad}_V(x_0)$, then for any $r \in [\lambda, \lambda')$, $\mathcal{L}_r$ is balanced with radius below $3/4$. Thus, we can let $V_0 = \mathcal{L}_r$, remove it from the graph, and repeat on the remaining part until we get a partition. □

References

[EEST08] Michael Elkin, Yuval Emek, Daniel A Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. *SIAM Journal on Computing*, 38(2):608–628, 2008.