Lower-Stretch Spanning Trees

Michael Elkin

Ben-Gurion University

Joint work with

Yuval Emek, Weizmann Institute

Dan Spielman, Yale University

Shang-Hua Teng, Boston University

The Setting

$G = (V, E, \omega)$ - a weighted undirected connected graph.

$dist_G(u, w)$ - the distance in G between vertices u and w is defined as the length of the lightest path connecting u and w in G.

Given a spanning tree T of G, and $(u, w) \in E$,

$$stretch_T(u, w) = \frac{dist_T(u, w)}{dist_G(u, w)} .$$

Total Stretch:

$$total\_stretch_T(E) = \sum_{(u,w) \in E} stretch_T(u, w) .$$

Average stretch:

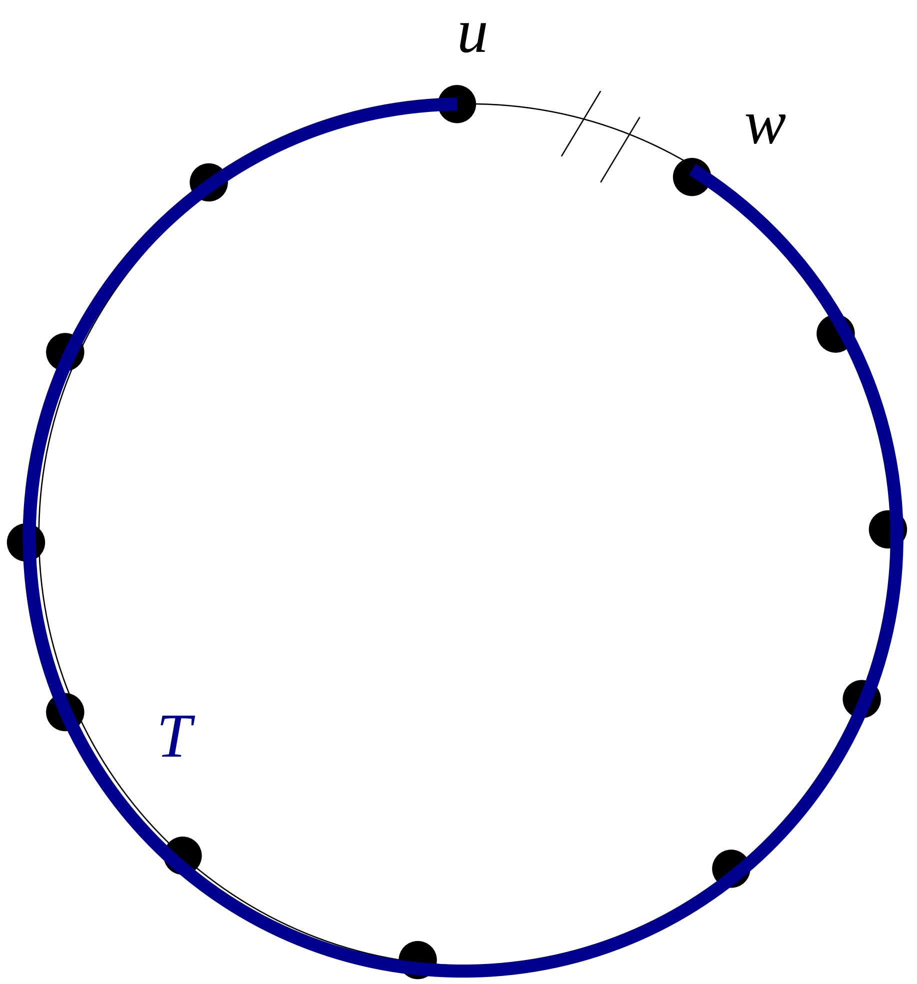
$$avg\_stretch_T(E) = \frac{total\_stretch_T(E)}{|E|} .$$

Irrelevance of Maximum Stretch

The notion of *maximum stretch*,

$$max\_stretch_T(E) = \max_{(u,w)\in E} \{stretch_T(u,w)\} ,$$

is not helpful in the context of trees,
as even for the unweighted n-cycle, any spanning tree has maximum stretch equal to $n-1$, but average stretch $2-2/n$.



Previous Work

The problem was introduced by [Alon,Karp,Peleg,West,92].

Thm (AKPW)
$\forall\ G = (V, E, \omega), |V| = n,\ \exists$ spanning tree $G' = (V, T, \omega)$, $T \subseteq E$, with

$$avg\_stretch_T(E) = exp(O(\sqrt{\log n \log\log n}))\ .$$

An alternative formulation (the dual problem): probabilistic spanning tree metric

$\forall\ G = (V, E, \omega),\ \exists$ a probability distribution $\mathcal{D}$ over a support set $\tau = \{T_1, T_2, \ldots, T_r\}$ of spanning trees of G s.t. for every edge $e \in E$,

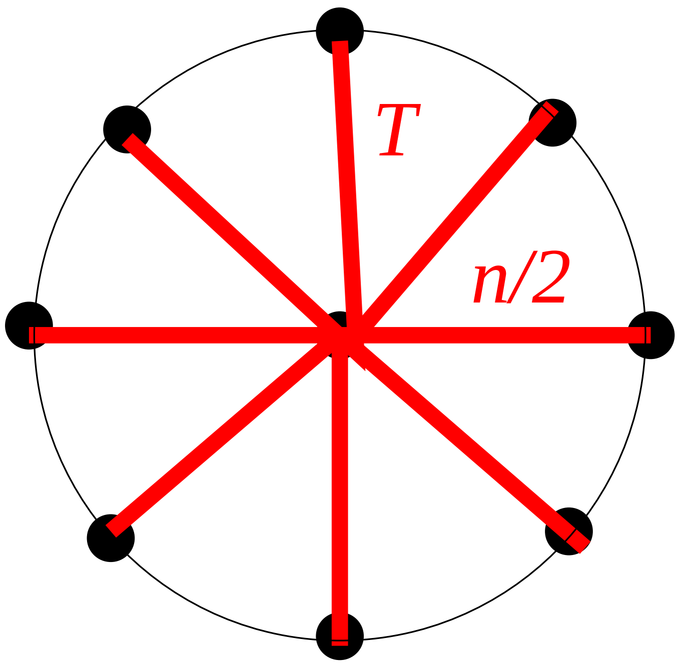
$$\mathbb{E}_{T \in_{\mathcal{D}} \tau}(stretch_T(e)) = exp(O(\sqrt{\log n \log\log n}))\ .$$

$$\Rightarrow \quad avg\_stretch \approx \mathbb{E}(max\_stretch)$$

Lower Bound: $\Omega(\log n)$.

Related work: virtual trees

A *virtual tree* T of G is allowed
to contain vertices or edges not present
in the original graph, but it is required that
(1) $\forall\ u, w \in V,\ \ dist_T(u, w) \geq dist_G(u, w)$;
(2) T spans V.



[Bartal,96]
Much tighter bounds on the average stretch and the expected maximum stretch for this setting, specifically, $O(\log^2 n)$.

[Bartal,98]
$O(\log n \log \log n)$.

[Fakchraenphol,Rao,Talwar,2003]
$O(\log n)$ - *tight !!*
([Bartal,96]: The l.b. of AKPW applies.)

Many applications for *on-line* and *approximation* algorithms.

Our result

An upper bound of $O(\log^2 n \log\log n)$ for the original problem of [Alon-Karp-Peleg-West], improving their bound of $exp(O(\sqrt{\log n \log\log n}))$.

The result applies also for *probabilistic spanning tree metrics*.

The gap is currently between $O(\log^2 n \log\log n)$ and $\Omega(\log n)$.

Applications of Our Result

1. Approximate solving of symmetric diagonally dominant linear systems.

   Running time $m \cdot \log(1/\epsilon) \cdot \log^{O(1)} n$ instead of $m \cdot \log(1/\epsilon) \cdot exp\{O(\sqrt{\log n \log\log n})\}$.

   Used when applying finite-element method to solve 2-dimensional elliptic PDEs.

2. Improving the upper bound on the value of the graph-theoretic game of AKPW (from $exp\{O(\sqrt{\log n \log\log n})\}$ to $O(\log^2 n \log\log n)$).

3. Improved approximation algorithm for the Minimum Communication Cost Tree Problem:

   This is a classic NP-hard problem, listed in [Garey,Johnson] book.

   Our result gives rise to the first poly-logarithmic approximation for this problem, improving the $exp\{O(\sqrt{\log n \log\log n})\}$-approximation of [Peleg,Reshef,98].

Overview of the Talk

1. Construction of spanning trees with stretch $O(\log^3 n)$ for *unweighted* graphs.

   (a) Introducing *star-decomposition*; stating the *-decomposition theorem.

   (b) Using *-decomposition to build low-stretch spanning trees.

   (c) Proving the *-decomposition theorem.

2. Improving the bound to $O(\log^2 n \log\log n)$.

3. Extending the bound to *weighted* graphs.

Construction of Trees with Stretch $O(\log^3 n)$

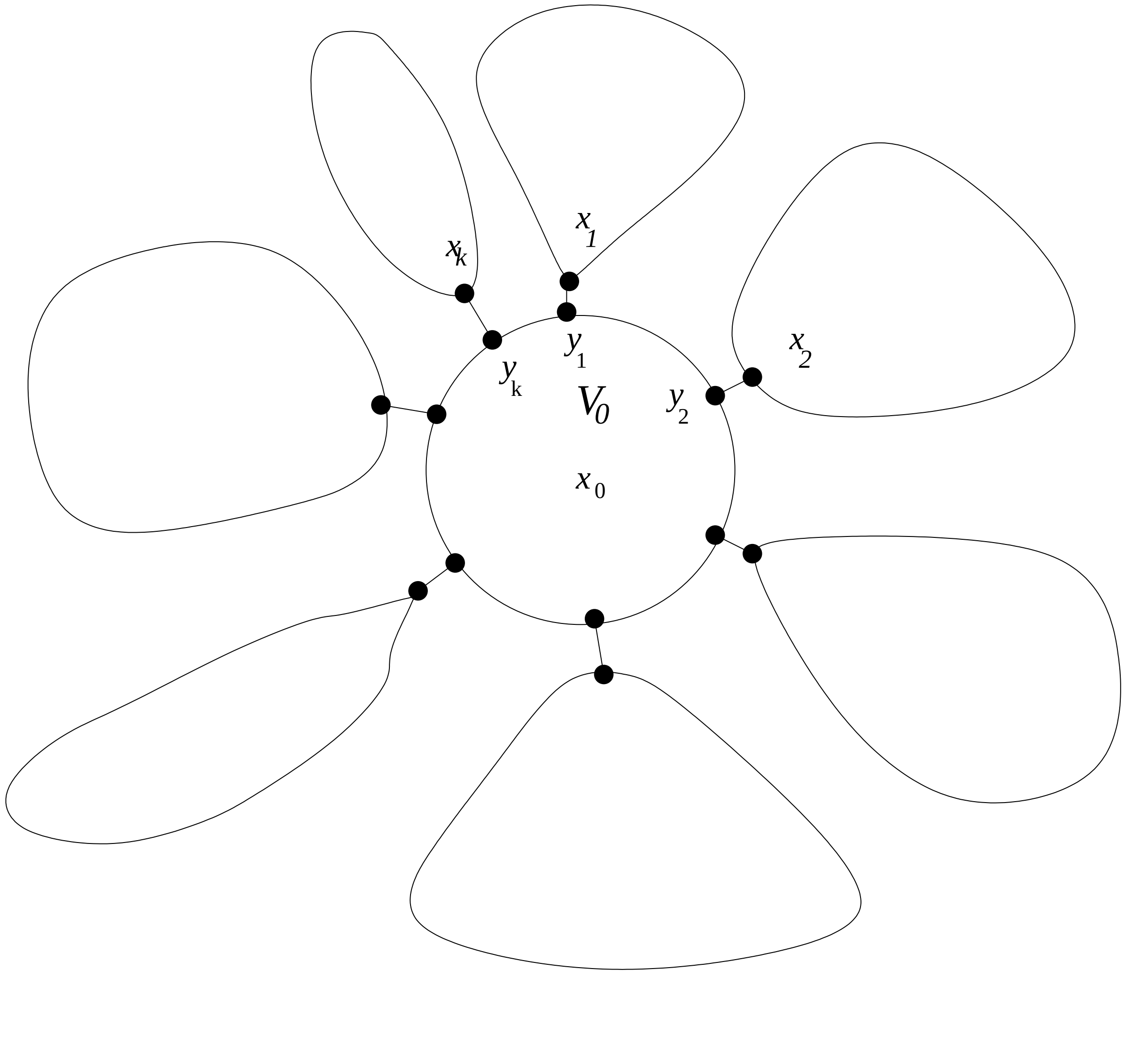
(For *unweighted* graphs.)

Star-Decomposition $(G = (V, E), x_0, \epsilon)$

Partition $\{V_0, V_1, \ldots, V_k\}$ of V
with center vertex $x_0 \in V_0$, $0 < \epsilon \leq 1/2$, and

1. $\forall\ i,\ 0 \leq i \leq k,\ G(V_i)$ is connected.

2. $\forall\ i,\ 0 \leq i \leq k,\ V_i$ contains an *anchor* x_i connected to $y_i \in V_0$ by an edge (x_i, y_i), called *bridge*.



x_k
x_1
x_2
y_k
y_1
y_2
V_0
x_0

*-Decomposition (cont.)

- $R/3 \leq r_0 \leq 2R/3$, where $R = Rad(G, x_0)$, and $r_i = Rad(G(V_i), x_i)$, for $i \geq 0$.

- $r_0 + 1 + r_i \leq (1 + \epsilon)R, \ \forall \ i \geq 1$.
  ($dist_G(y_i, x_i) = 1$ for unweighted graphs).

  $\max\{r_0 + dist_G(y_i, x_i) + r_i \mid i \geq 1\}$ is the radius wrt x_0 of the graph induced by the *-decomposition.

  $\forall i \geq 1, \quad r_i \leq (2/3 + \epsilon)R$.

The *-Decomposition THM

$\partial(V_0, V_1, \ldots, V_k)$ - the set of edges crossing between different components.

The *cost* of *-decomposition is the number of edges in $\partial(V_0, V_1, \ldots, V_k)$.

THM: $\forall\ G = (V, E),\ x_0 \in V,\ 0 < \epsilon \leq 1/2$,
$\exists$ *-decomposition
$(\{V_0, V_1, \ldots, V_k\}, (x_0, x_1, \ldots, x_k), (y_1, \ldots, y_k))$
with cost $O((m \log m)/(\epsilon \cdot R))$,
where $m = |E|,\ R = Rad(G, x_0)$.

Intuitively, only an $\frac{1}{R}$-fraction of edges of the graph cross between different components.

The Construction

(of $O(\log^3 n)$-stretch spanning trees based on the *-decomposition theorem.)

Procedure $ST(G, x_0)$:

If $Rad(G, x_0) \geq \log n$ then:

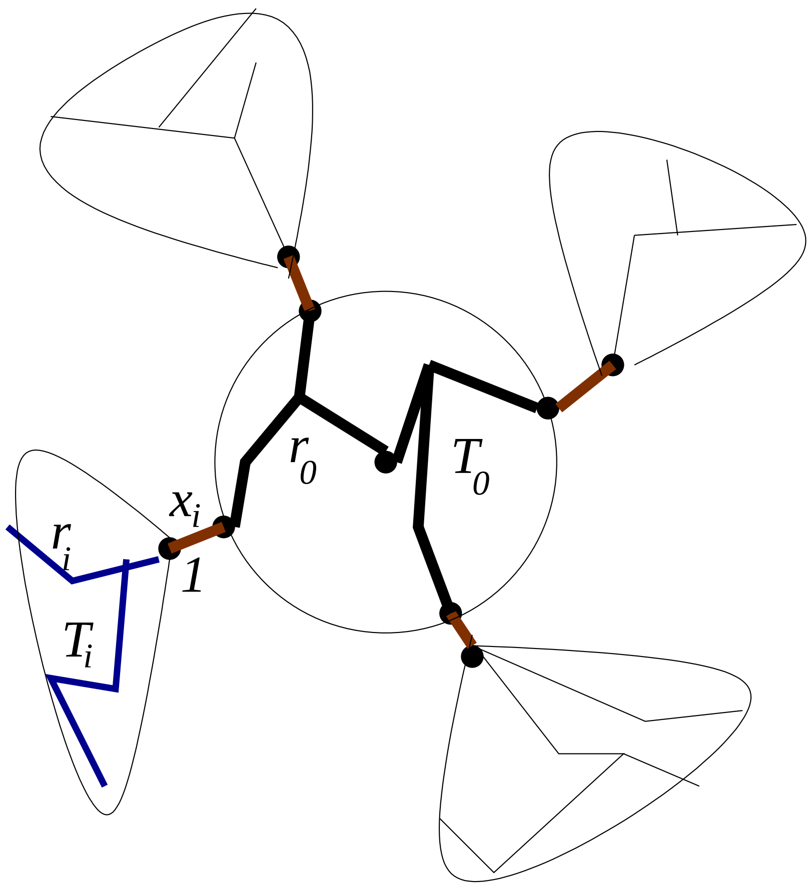
1. $(V_0, V_1, \ldots, V_k, \vec{x}, \vec{y}) =$
   $StarDecomp(G, x_0, \epsilon = 1/\log n)$;

2. /* Recurse on each component building a tree rooted in the respective anchor. */
   For $i = 0, 1, \ldots, k$ do
   $T_i = ST(G(V_i), x_i)$;

3. Return$(T = (\bigcup_i T_i) \cup \bigcup_i \{(y_i, x_i)\})$;

Else

Return$(T = BFS(G, x_0))$;



Analysis: Radius

For $t \geq 0$,
$K_t(G)$ is the graph that one would obtain if the algorithm were forced to return output after t levels of recursion.

$K_0(G) = G$.

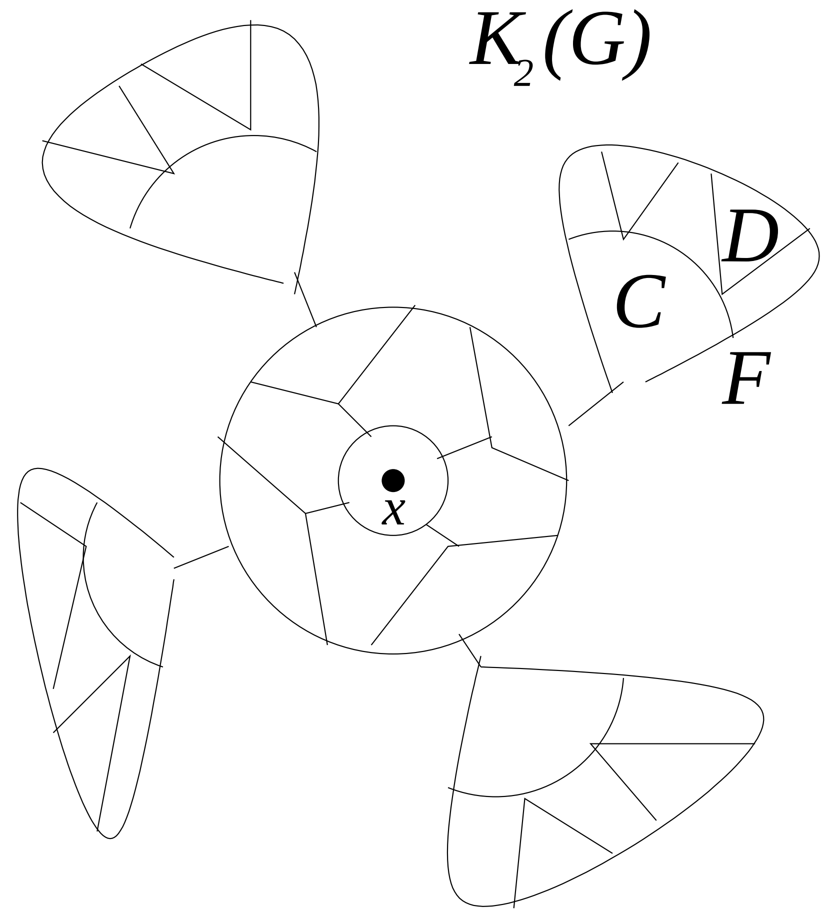
$Rad(K_1(G), x_0) \leq (1 + \epsilon) \cdot Rad(K_0(G), x_0)$.

Consequently,
$Rad(K_t(G), x_0) \leq (1 + \epsilon)^t \cdot Rad(K_0(G), x_0)$.

Since the radii of components decrease by a constant factor $(2/3 + \epsilon)$ on each level of the recursion, it follows that $t = O(\log n)$.

Set $\epsilon = \frac{1}{\log n}$.
Then $Rad(T, x_0) = O(1) \cdot Rad(G, x_0)$,

where T is the resulting tree.



Analysis: Stretch

The contribution of edges crossing between different components of the *-decomposition on the 1st level of the recursion to $total\_stretch_T(E)$ is:

$$\sum_{(u,w)\in\partial(V_0,\dots,V_k)} stretch_T(u,w) = \sum_{(u,w)\in\partial(V_0,\dots,V_k)} dist_T(u,w) \leq \sum_{(u,w)\in\partial(V_0,\dots,V_k)} 2\cdot Rad(T,x_0) \leq \sum_{(u,w)\in\partial(V_0,\dots,V_k)} O(1)\cdot Rad(G,x_0) =$$

$$O(|\partial(V_0,V_1,\dots,V_k)|\cdot R) \leq$$

$$O\left(\frac{m\log m}{R\epsilon}\cdot R\right) = O(m\log^2 m)\ .$$

On the 2nd level of the recursion:

$$\sum_{i=0}^{k} \sum_{(u,w)\in\partial(V_{i0},V_{i1},\ldots,V_{i\ell})} \mathit{stretch}_T(u,w) \le$$

$$\sum_{i=0}^{k} \sum_{(u,w)\in\partial(V_{i0},V_{i1},\ldots,V_{i\ell})} O(1)\cdot \mathit{Rad}(G(V_i),x_i) =$$

$$O\left(\sum_{i=0}^{k} |\partial(V_{i0},V_{i1},\ldots,V_{i\ell})|\cdot r_i\right) =$$

$$O\left(\sum_{i=0}^{k} \frac{|E(V_i)|\cdot \log m}{\epsilon\cdot r_i}\cdot r_i\right) =$$

$$O(\log^2 m)\cdot \sum_{i=0}^{k} |E(V_i)| = O(m\cdot \log^2 m) .$$

Since there are $O(\log n)$ recursion levels, the total stretch of the edges cut on some recursion level is $O(m \log^3 m) = O(m \log^3 n)$.

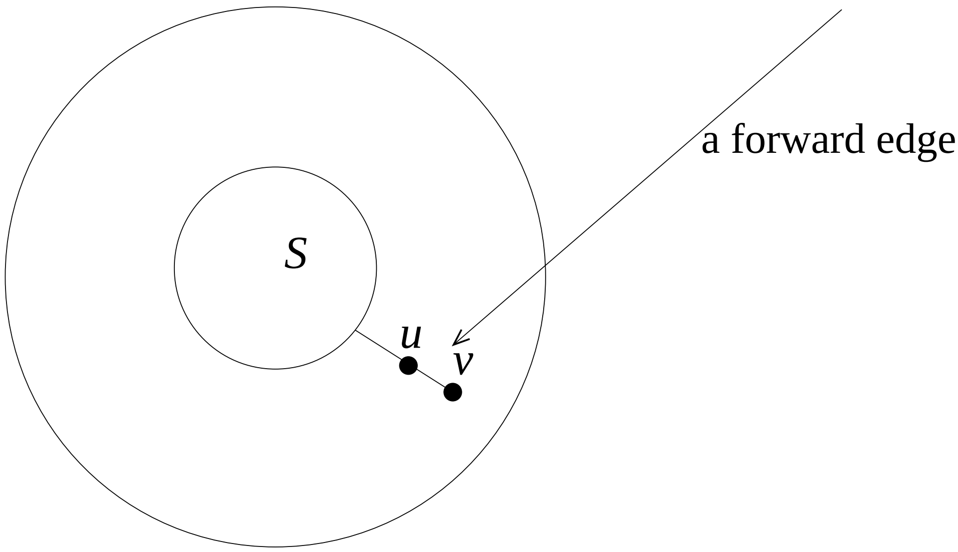
In addition, there are edges that survive all the way to the bottom level of the recursion.
Each of them has stretch $O(\log n)$.
They contribute $O(m \cdot \log n)$ to the total stretch.

$$\Rightarrow \mathit{total\_stretch}_T(E) = O(m \cdot \log^3 n)$$
$$\Rightarrow \mathit{avg\_stretch}_T(E) = O(\log^3 n) \ .$$

Constructing *-Decomposition: Forward Edges

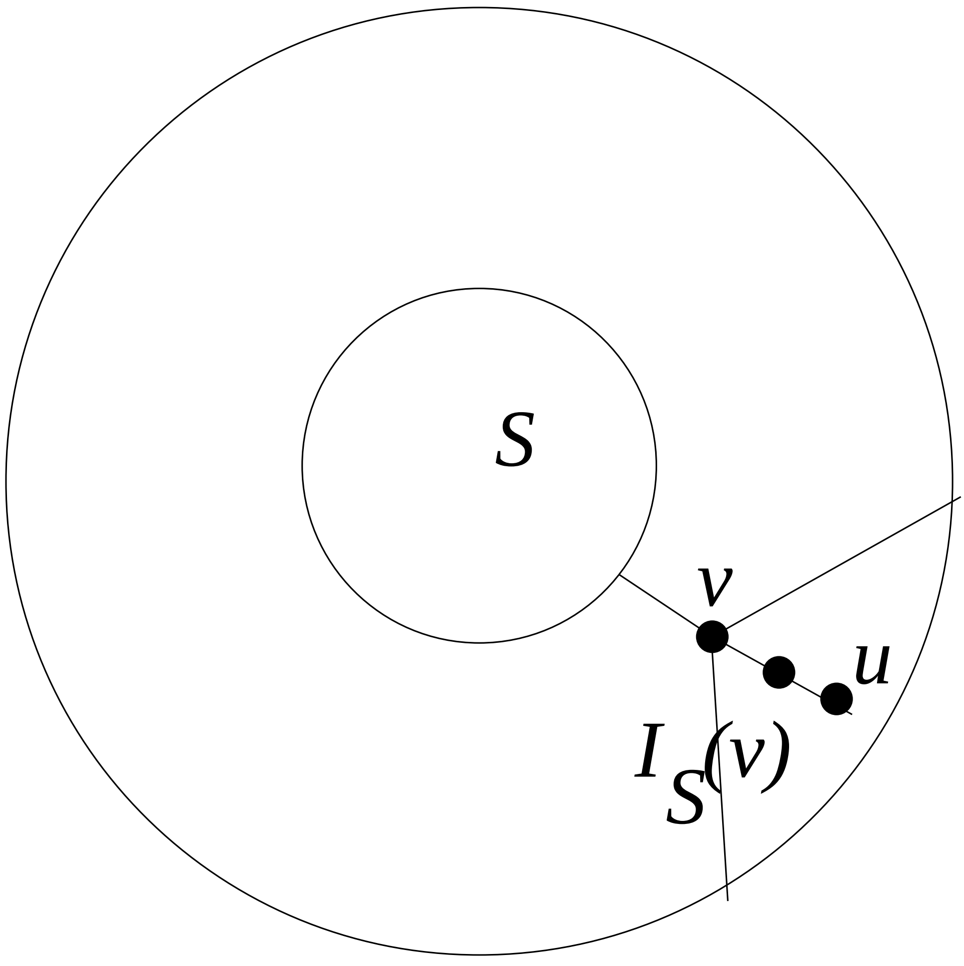
Forward Edges: Given $V_0 = S \subseteq V$, the set of *forward edges* induced by S is

$$F(S) = \{(u \to v) \mid (u,v) \in E, \; dist(S,u) + 1 = dist(S,v)\} \; .$$



Constructing *-Decomposition: Ideals

For a vertex $v \in V$,
the *ideal* $I_S(v)$ of v induced by S,
is the set of all vertices
reachable from v by directed forward edges,
including v.



Constructing *-Decomposition: Cones

For a vertex $v \in V$, the *cone* of width ℓ around v induced by S, denoted $C_S(\ell, v)$, is the set of vertices reachable from v by a path containing at most ℓ non-forward edges (that are not in $F(S)$).

$(I_S(v) = C_S(0, v))$

*-Decomposition

- Cut a ball S of radius roughly $R/3$ around x_0 such that the number of edges in the cut is $O((m \log n)/R)$.
  Set $V_0 = S$.

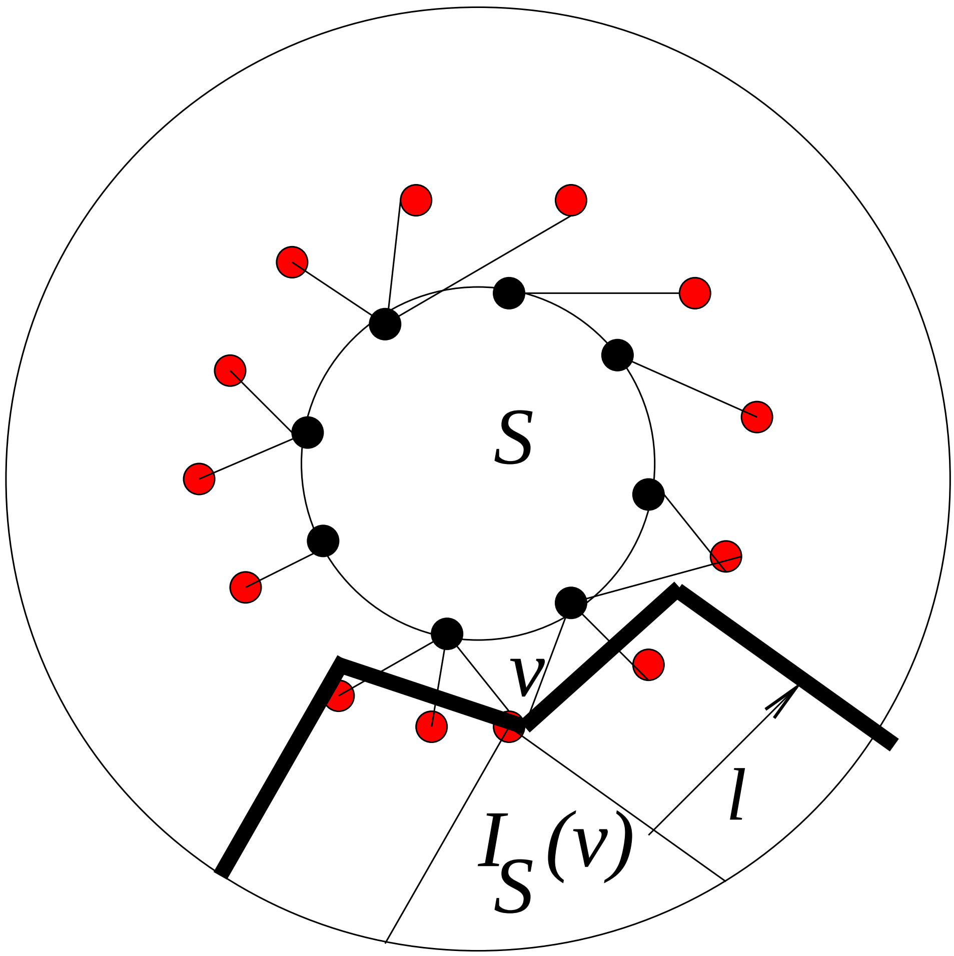
- Pick an arbitrary vertex v on the ball-shell of S (*ball-shell* - the set of neighbors of vertices of S that do not belong to S).
  Build an ideal $I_S(v)$,
  and grow a cone $C_S(v, \leq \frac{R}{\log n})$ with width at most $\frac{R}{\log n}$.
  The number of edges cut is
  $\leq \frac{\log^2 n}{R} \cdot |E(C_S)|$.

- Iterate until all vertices of the ball-shell are exhausted.

  Let $\{V_1, V_2, \ldots, V_k\}$ be the set of cones.

  The number of edges that belong to the cone cuts is $\sum_{C \in Cones} \frac{\log^2 n}{R} \cdot |E(C)| = O(\frac{m \cdot \log^2 n}{R})$.

THM: $\forall\ G = (V, E),\ x_0 \in V,\ 0 < \epsilon \leq 1/2$, $\exists$ *-decomposition $(\{V_0, V_1, \ldots, V_k\}, (x_0, x_1, \ldots, x_k), (y_1, \ldots, y_k))$ with cost $O((m \log m)/(\epsilon \cdot r))$, where $m = |E|$.



Red points = Ball−shell of S

Constructing *-Decomposition: Properties

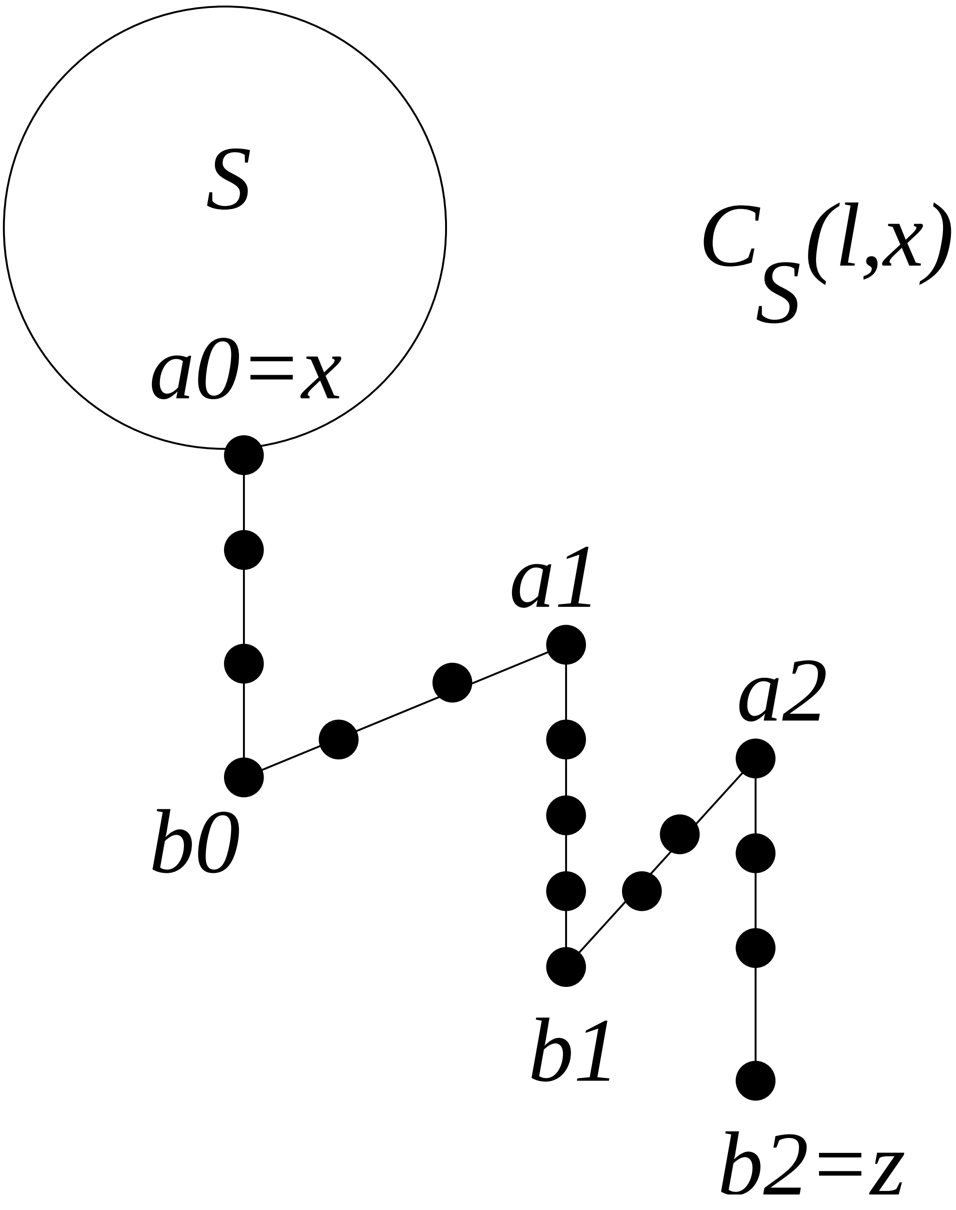
Lm: Let $\psi = \max\{dist_G(S,v) \mid v \in V\}$.
For every $x \in S$, and $\ell \geq 0$
$Rad(C_S(\ell,x),x) \ \leq \ \psi + 2\ell$.

Intuition for the proof:

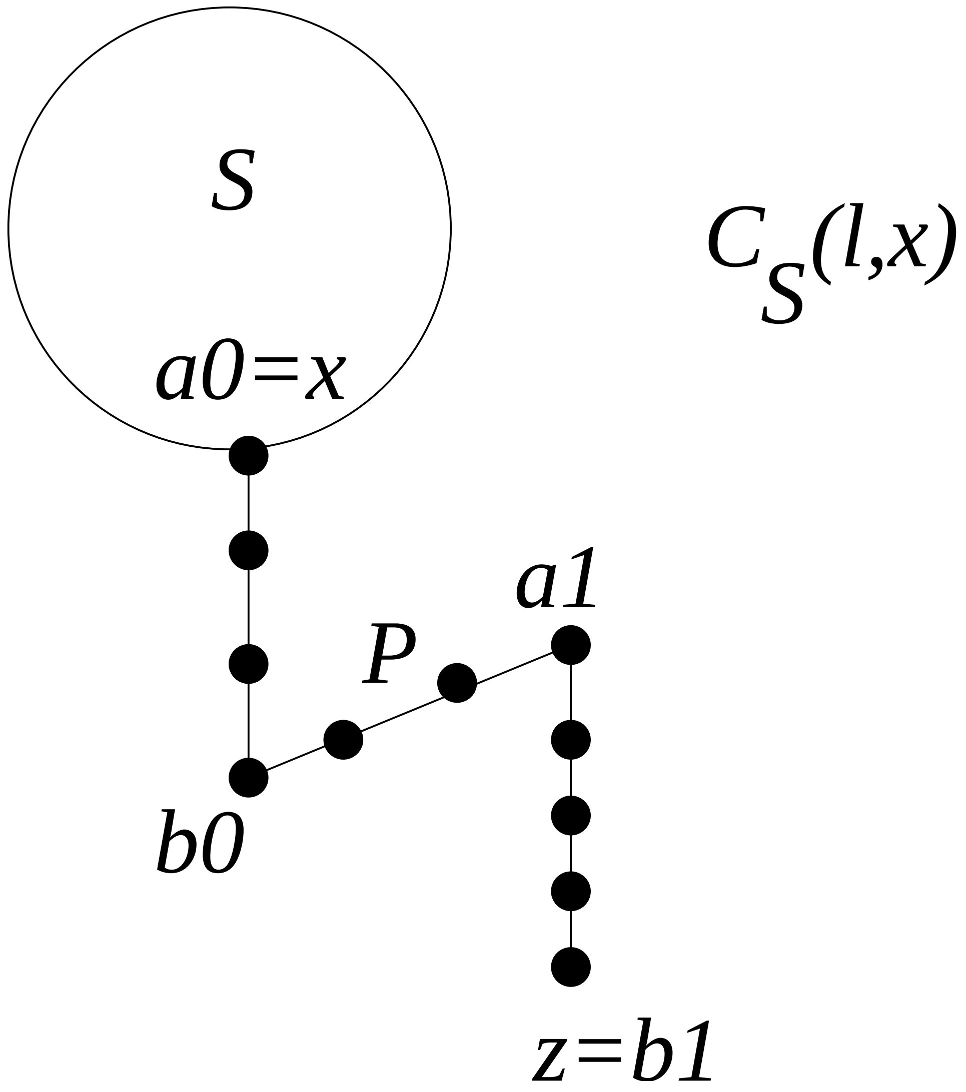
$z \in C_S(\ell,x)$.
$P = P(x,z)$ is the shortest x-z-path
with $\leq \ell$ non-forward edges.
If $P \subseteq F(S)$ then $|P| \leq \psi$.

Each non-forward edge is
(in the worst-case) one step backwards.

Each of the ℓ steps backwards
needs to be compensated by a step forward.


S
$C_S(l,x)$
a0=x
a1
a2
b0
b1
b2=z

A *simplistic* path P



The Proof of the Lemma

Assume that P is a simplistic path:
$P = (x = a_0, \ldots, b_0, \ldots, a_1, \ldots, b_1 = z)$.

$$
\begin{aligned}
dist(S, b_0) &= dist(a_0, b_0), \\
dist(S, a_1) &\geq dist(S, b_0) - \ell, \\
dist(S, b_1) &= dist(S, a_1) + dist(a_1, b_1), \\
\psi \geq dist(S, z) &= dist(S, b_1) \geq \\
&\geq dist(S, b_0) + dist(a_1, b_1) - \ell,
\end{aligned}
$$

Hence the #forward edges in P is

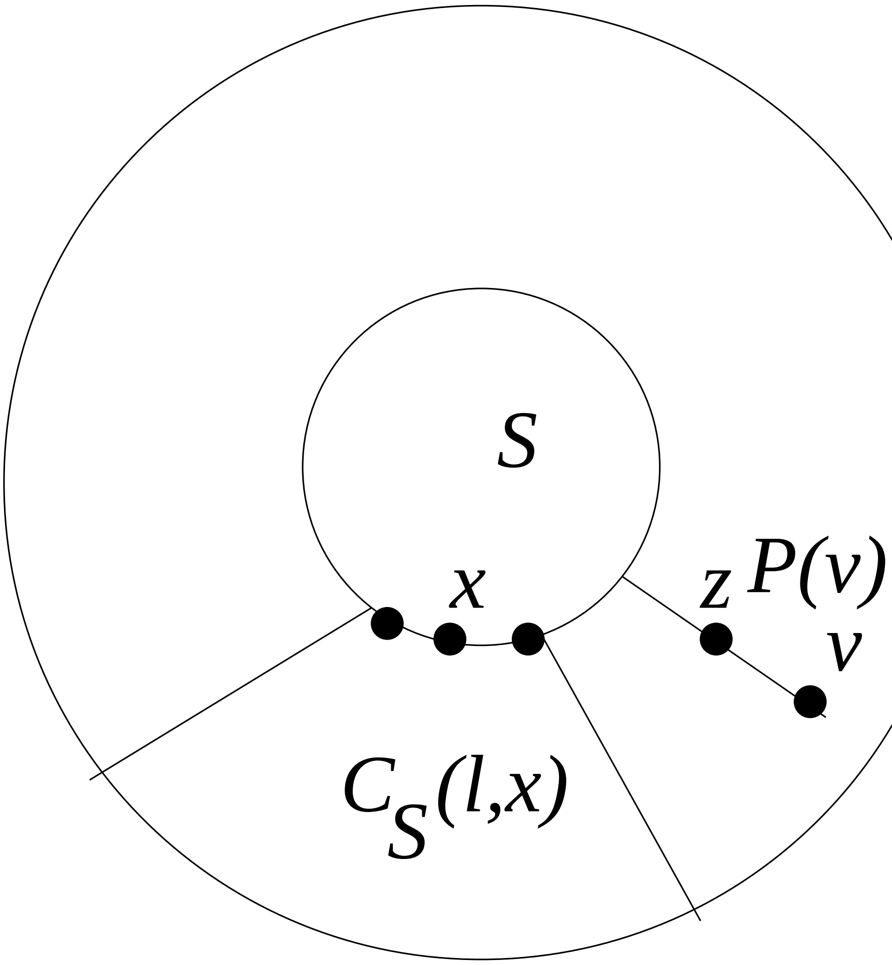
$$
\begin{aligned}
dist(a_0, b_0) + dist(a_1, b_1) &\leq \psi + \ell, \\
|P| &\leq dist(a_0, b_0) + dist(a_1, b_1) + dist(b_0, a_1) \\
&\leq (\psi + \ell) + \ell = \psi + 2\ell.
\end{aligned}
$$

QED

Deleting Cones

Lm: For a set $S \subseteq V$, a vertex $x \in S$, and $\ell \geq 0$, let $V' = V \setminus C_S(\ell, x)$, $S' = S \setminus C_S(\ell, x)$, and $\psi = \max\{dist_G(S, v) \mid v \in V\}$.

Then
$\max\{dist_{G(V')}(S', v) \mid v \in V'\} \leq \psi$.



Proof:

$P(v)$ - the shortest path between S and v in G.

All edges of $P(v)$ are forward edges.

If $P(v) \cap C_S(\ell, x) \neq \emptyset$ then
$v \in C_S(\ell, x)$.

Hence for $v \in V' = V \setminus C_S(\ell, x)$,
$P(v)$ lies entirely in V'.

$dist_{G(V')}(S', v) \leq |P(v)| \leq \psi$.

QED

Cutting a Ball

[Awerbuch,Peleg,90],[Leighton,Rao,88]

Start with a vertex $v \in V$,
and set $S = \{v\}$, and $\delta = \frac{3 \cdot \log m}{R}$,
for $R = Rad(G, v)$.

On every iteration test whether

(1) $|E(\Gamma(S))| \geq (1 + \delta) \cdot |E(S)|$,

where $\Gamma(S) = \{y \mid \exists x \in S,\ (x, y) \in E\} \cup S$.

If this condition holds, set $S = \Gamma(S)$.

Iterate until condition (1) is violated.

Return(S).

Cutting a Ball: Analysis

Radius r of S satisfies:

$(1 + \delta)^{r+1} \leq m.$

Hence $r \leq \frac{\log m}{\delta} - 1 \leq R/3.$

The number of edges in the cut is

$|E(S, \Gamma(S) \setminus S)| \leq \delta |E(S)| = O(\frac{\log n}{R} |E(S)|).$

To build a ball of radius
between $R/3$ and $2R/3$, we start with
$S = \{z \mid dist_G(v, z) \leq R/3\}$,
for some vertex v,
and use the same procedure.

Cutting Cones

The same technique applies to cutting cones.

To keep the radius growth in check, use $\delta = \frac{O(\log^2 n)}{R}$.

Hence the width of each cone is $O(\frac{\log m}{\delta}) = O(R/\log n)$.
(The width of a cone is analogous to the radius of a ball.)

The number of edges in cuts between different cones is

$$\sum_{C \in Cones} |E(C, \Gamma(C) \setminus C)| \leq$$
$$\delta \cdot \sum_{C \in Cones} |E(C)| \leq$$
$$\delta m = O\left(\frac{\log^2 n}{R} \cdot m\right) .$$

Hence the *-decomposition's cost is $O\left(\frac{\log^2 n}{R} \cdot m\right)$.

Other ways of Cutting

Using [Linial,Saks, 92] (a.k.a [Bartal, 96]) cutting technique one can achieve similar parameters, and randomize the construction.

In other words, this way one obtains a probability distribution over spanning trees with expected stretch of $O(\log^3 n)$ for *every* edge.

Radius is chosen from the exponential distribution defined by the density function
$p(x) = \lambda e^{-\lambda x}$
with λ being $O(\frac{1}{R})$ for a ball,
and $\frac{\log n}{R}$ for each cone.

Improving the Bound

(to $O(\log^2 n \log\log n)$)

The idea:

If $|E(C)|$ of a cone C is at most $\frac{m}{2^{\sqrt{\log m}}}$, then we "saved" $\sqrt{\log m}$ steps of recursion.

Otherwise, we grow the cone again with radius $r = \frac{R}{\log n}$.
In this case the fraction δ of edges that are cut satisfies $|E(C)| \cdot (1+\delta)^r \leq |E|$.
Hence

$$\delta \leq \frac{\log(|E|/|E(C)|)}{r} \leq \frac{\sqrt{\log m}}{r} = O\left(\frac{\log^{3/2} n}{R}\right).$$

In either case we save a factor of $\sqrt{\log n}$.

Improving the Bound (cont.)

More formally:

Grow the cone C to radius $r \leq \frac{1}{2} \cdot \frac{R}{\log n}$.

1. If $|E(C)| \leq \frac{m}{2\sqrt{\log m}}$, leave C as is, and turn to constructing the next cone.

2. Otherwise, if $|E(C)| > \frac{m}{2\sqrt{\log m}}$, grow C again to radius $r \leq \frac{1}{2} \cdot \frac{R}{\log n}$.

   In this case the number of edges in the cut is

$$|\partial(C)| \leq O\left(\frac{\log^{3/2} n}{R}\right) |E(C)|,$$

   saving a factor of $\sqrt{\log n}$.

Improving the Bound (cont.)

In case 1 the fraction $\frac{|\partial(C)|}{|E(C)|}$
can be as large as $\frac{O(\log^2 n)}{R}$,
but when summing up over all recursion levels,
each edge e can participate
in at most $O(\sqrt{\log m})$ cones
(on different recursion levels)
on which case 1 occurs.

(Because on these levels the size
of the respective cone shrinks significantly.)

Altogether, we pay $O(\frac{\log^{2.5} n}{R} \cdot m)$
in both cases, while maintaining the limited
growth of the radius (from R to $R(1 + \frac{1}{\log n})$
on each recursion level).

Repetitive Growing

Instead of doing this just twice,
we can grow the cones repetitively
t times, for an arbitrary parameter t.

The only problem with increasing t
is that to keep the radius growth as small
as previously, we have to increase the radius
from R to $R(1 + \frac{1}{t} \cdot \frac{1}{\log n})$
on each recursion level.

For $t = 3$:
Grow the cone C to radius
$r \leq \frac{1}{3} \cdot \frac{R}{\log n}$.

1. If $|E(C)| \leq \frac{m}{2^{\log^{2/3} m}}$, leave C as is.
   There can be at most $\log^{1/3} m$ recursion levels of this kind,
   saving a factor of $\log^{2/3} m$.

2. If $|E(C)| > \frac{m}{2^{\log^{2/3} m}}$,
   grow it again to radius r,
   $r \leq \frac{1}{3} \cdot \frac{R}{\log n}$.

   If $|E(C)| < \frac{m}{2^{\log^{1/3} m}}$ then

$$\delta \ \leq \frac{\log(|E|/|E(C)|)}{r} < \frac{\log^{2/3} m}{r} \ ,$$

   and so

   (a) $|\partial(C)| \leq |E(C)| \cdot \frac{\log^{2/3} m}{r}$, saving a factor of $\log^{1/3} m$.

   (b) There can be at most $\log^{2/3} m$ recursion levels of this kind, saving another factor of $\log^{1/3} m$.

3. Else if $|E(C)| > \frac{m}{2^{\log^{1/3} m}}$ then
   grow C once again to radius r,
   $r \leq \frac{1}{3} \cdot \frac{R}{\log n}$.

In this case $\delta \leq \frac{\log(|E|/|E(C)|)}{r} \leq \frac{\log^{1/3} m}{r}$, and

$$|\partial(C)| \leq |E(C)| \cdot \frac{\log^{1/3} m}{r} ,$$

saving a factor of $\log^{2/3} m$.

In all the three cases the radius
grows from R to $R(1 + \frac{1}{\log n})$,
as required.

Best choice of t
(the number of steps of repetitive growing)
is $t = \Theta(\log \log n)$.
This gives rise to the bound of
$O(\log^2 n \log \log n)$.

Extending the Argument to Weighted Graphs

Extending the cutting lemma:

Fix a vertex v. For $r \geq 0$, let
B_r be a ball of radius r around v.
Let $\partial(B_r) = \{e = (u, w) \mid u \in B_r, w \notin B_r\}$,
$Vol(B_r) = |E(B_r \cup \partial(B_r))|$, and

$$cost(\partial(B_r)) = \sum_{e \in \partial(B_r)} \frac{1}{length(e)} .$$

The intuition for $cost(\partial(B_r))$:
our analysis provides an upper bound L for the distance between u and w in the resulting tree for every edge $e = (u, w)$ of a certain cut $\partial(B_r)$.
The contribution of all edges of this cut to the total stretch is at most

$$\sum_{e \in \partial(B_r)} \frac{L}{length(e)} = L \cdot cost(\partial(B_r)) .$$

Lm: $\forall \lambda, \lambda',\ 0 \leq \lambda < \lambda'$,
$\exists r \in [\lambda, \lambda')$ s.t.

$$cost(\partial(B_r)) \leq \frac{Vol(B_r)}{\lambda' - \lambda} \cdot \log\left(\frac{m}{Vol(B_\lambda)}\right) .$$

In the unweighted case:

$cost(\partial(B_r)) = |E(\partial(B_r))|$,
$\lambda = R/3,\ \lambda' = 2R/3$.

The Proof of the Weighted Cutting Lemma

Let $r_1 \leq r_2 \leq \dots r_{n-1}$
be the distances of all the vertices
$v_1, v_2, \dots, v_{n-1}$ from v (their "norms").

Assume wlog that

$$dist_G(v_i, v_j) = |r_j - r_i| \ .$$

The generality is not lost here
because $dist_G(v_i, v_j) \geq |r_j - r_i|$,
and decreasing these distances
does not change $Vol(B_r)$, but may increase

$$cost(\partial(B_r)) = \sum_{e \in \partial(B_r)} \frac{1}{length(e)} \ .$$

A crucial quantity:

$$\mu_i = |E(B_i)| + \sum_{(v_j, v_k) \in E | j \leq i < k} \frac{r_i - r_j}{r_k - r_j} \ .$$

(For every edge that crosses the boundary of B_r, we account for its portion inside B_r.)

It follows that

$$\begin{aligned} \mu_{i+1} &= \mu_i + cost(\partial(B_i)) \cdot (r_{i+1} - r_i) \ , \\ |E(B_i)| \ \le \ \mu_i \ &\le \ Vol(B_i) \ = \ |E(B_i \cap \partial(B_i))| \ . \end{aligned}$$

Let a, b be the indices s.t. $r_{a-1} \le \lambda < r_a$, $r_b < \lambda' \le r_{b+1}$.

Trivial cases:

1. $b < a$

   In this case no vertex has norm between λ and λ'.

   Choose r between them, and each edge crossing the rth latitude contributes to $cost(\partial(B_r))$ at most $\frac{1}{length} \le \frac{1}{\lambda' - \lambda}$, and thus $\partial(B_r) \le \frac{Vol(\partial(B_r))}{\lambda' - \lambda} \le \frac{Vol(B_r)}{\lambda' - \lambda}$.

2. $\exists i \in [a-1, b]$ s.t.

$$r_{i+1} - r_i \;\geq\; \frac{\lambda' - \lambda}{\log \frac{m}{Vol(B_\lambda)}} \;.$$

Analogously, choose r between r_i and r_{i+1}, and each edge contributes to $cost(\partial(B_r))$ at most $\frac{1}{r_{i+1}-r_i}$.

The non-trivial case:
$[a, b] \neq \emptyset$, and $\forall i \in [a-1, b]$

$$r_{i+1} - r_i \;<\; \frac{\lambda' - \lambda}{\eta} \;,$$

where $\eta = \log \frac{m}{Vol(B_\lambda)}$.

By the choice of a, $Vol(B_\lambda) = Vol(B_{a-1})$, and so $\eta = \log \frac{m}{Vol(B_{a-1})}$.

Assume for contradiction that $cost(\partial(B_i)) > \mu_i \cdot \frac{\eta}{\lambda'-\lambda}$, for all $i \in [a-1, b]$.

Then

$$
\begin{aligned}
\mu_{i+1} &= \mu_i + cost(\partial(B_i))(r_{i+1} - r_i) \\
&> \mu_i + \mu_i \frac{\eta}{\lambda' - \lambda}(r_{i+1} - r_i) \\
&= \mu_i\left(1 + \frac{\eta}{\lambda' - \lambda}(r_{i+1} - r_i)\right) .
\end{aligned}
$$

Hence

$$
\begin{aligned}
\mu_{b+1} &> \mu_{a-1} \cdot \Pi_{i=a-1}^{b}\left(1 + \frac{\eta}{\lambda' - \lambda}(r_{i+1} - r_i)\right) \\
&\geq \mu_{a-1} \cdot 2^{\sum_{i=a-1}^{b} \frac{\eta}{\lambda' - \lambda}(r_{i+1} - r_i)} \\
&= \mu_{a-1} \cdot 2^{\frac{\eta}{\lambda' - \lambda}(r_{b+1} - r_{a-1})} \\
&\geq \mu_{a-1} \cdot 2^{\eta} \\
&= \mu_{a-1} \cdot \frac{m}{|E(B_{a-1})|} \\
&\geq m \ .
\end{aligned}
$$

This is a contradiction.

Extension to the Weighted Case: Using Weighted Cutting

A direct application of the lemma results in a bound of $O(\log^2 n \cdot \log Rad(G))$.

To improve, contract all edges of length at most $\frac{Rad(G)}{n+1}$, and uncontract them on the lower levels of the recursion, when $Rad(C)$ becomes smaller (where C is the cluster on which the recursion is invoked).

This way each edge appears on $O(\log n)$ recursion levels, and contributes $O(\log^3 n)$ to the total stretch.

Summary

- Proved that for every undirected (possibly weighted multi-)graph there exists a spanning tree with a polylogarithmic average stretch.

  Specifically, our bound is $O(\log^2 n \log\log n)$.

- The same applies to *probabilistic spanning tree metrics*.

- The construction requires running time $O(m \log n + n \log^2 n)$.

- Applications to efficient solving of symmetric diagonally-dominant linear systems.

- Introduced a novel graph decomposition technique (**-decomposition*).

Open Problems

Close the gap between the upper and lower bounds.

A simpler problem: for *planar unweighted* graphs.

Stronger l.b.s for AverageStretch + AlmostBFS?

Recent progress:
[Emek,Peleg,2006]
Proved a logarithmic upper bound for *series-parallel unweighted* graphs.