601.435/635 Approximation Algorithms **Lecturer:** Michael Dinitz
Topic: Local Search: Min-Degree Spanning Tree **Date:** 2/11/21
Scribe: Michael Dinitz

6.1 Problem Definition

Today we'll be studying the following, problem, known as Minimum Degree Spanning Tree (MDST). Given a graph $G = (V, E)$, we will let $d_G(u)$ be the degree of u in G and will let $\Delta(G) = \max_{v \in V} d_G(v)$.

Input: Connected graph $G = (V, E)$.

Feasible solution: Spanning tree T of G.

Objective: $\min \Delta(T)$.

It's not hard to prove that MDST is NP-hard via a reduction from Hamiltonian Path (this might be useful for your homework!).

6.2 Algorithm: Attempt 1

Recall that the fundamental cycle of a non-tree edge e (with respect to some spanning tree) is the unique cycle created by adding e to the spanning tree. This allows us to make an obvious definition of a local move: it will be a pair (e, e'), where e is a non-tree edge and e' is a tree edge on the fundamental cycle of e (the cycle created in the tree by adding e). Clearly this maintains the feasibility of the solution (we went from one spanning tree to a different one). So it suggests the following local search algorithm:

Algorithm 1 Local Search 1 for Min-Degree Spanning Tree

Input: A graph $G = (V, E)$
Output: A spanning tree T

```
Find a spanning tree T of G
while there is a local move which decreases the max degree of the current T do
    do the move
end while
Output T
```

Unfortunately, as shown in Figure 6.2.1, the algorithm may not work well. There is a local optimum with maximum degree d as shown in the left figure (each non-leaf node in the tree has degree d) while the OPT is 3 as shown in the right figure.

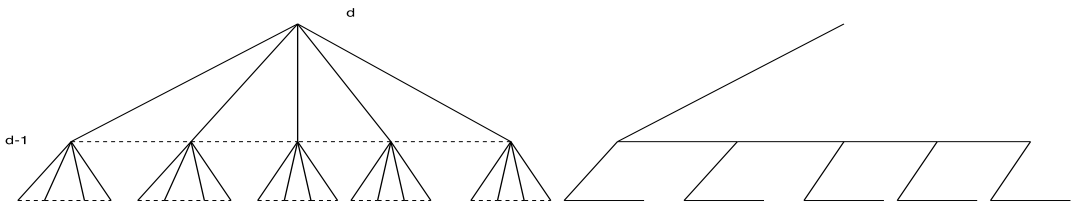

Figure 6.2.1: Bad instance for local search algorithm

6.3 Algorithm: Attempt 2

So we have to change our algorithm. Intuitively, one reason we got stuck earlier is that there were not any moves which improved the global max degree. On the other hand, there *were* moves that improved degrees *locally*. Consider, for example, some node u who is a child of the root in the previous example. If we add one of the edges between two of its children $\{v, w\}$, then we can remove the edge $\{u, v\}$, thus decreasing the degree of u while increasing the degree of a node who had extremely low degree. So this seems like a "win". This suggests the following definition of an *improving* local move.

Definition 6.3.1 *Let $u \in V$. Then $(u, \{v, w\})$ is a* u-improvement *if there is an edge $\{u, x\}$ on the fundamental cycle of $\{v, w\}$ so that we can swap $\{v, w\}$ for $\{u, x\}$ to get T' so that* $\max\{d_{T'}(v), d_{T'}(w)\} \leq d_{T'}(u) = d_T(u) - 1$

This suggests an obvious algorithm: keep finding u-improvements until there are no such improvements, for all $u \in V$. However, to the best of my knowledge, it's still an open question whether this algorithm terminates in polynomial time. So instead, we'll only try to improve nodes which are "close" to the maximum degree. This gives the following algorithm:

Algorithm 2 Local Search 2 for Min-Degree Spanning Tree

Input: A graph $G = (V, E)$
Output: A spanning tree T

```
Find a spanning tree T of G
while There is u-improvement with d_T(u) ≥ Δ(T) − log n for the current T do
    do the improvement
end while
Output T
```

6.3.1 Time Complexity

Theorem 6.3.2 *The running time of the algorithm is polynomial.*

Proof: The proof works by analyzing a potential function. Let $\Phi(v) = 3^{d_T(v)}$, and let $\Phi(T) = \sum_{v \in V} \Phi(v) = \sum_{v \in V} 3^{d_T(v)}$.

First note that $\Phi(T) \geq \sum_{v \in V} 3 = 3n$, since all nodes have degree at least 1 in T. Similarly, $\Phi(T) \leq n \cdot 3^n$ since every node has degree at most n. We now show that the potential function drops notably in each iteration.

Claim 6.3.3 *Suppose we make a u-improvement $(u, \{v, w\})$ with $d_T(u) \geq \Delta(T) - \log n$, obtaining a new spanning tree T'. Then $\Phi(T') \leq (1 - \frac{2}{9n^3})\Phi(T)$.*

Proof: Suppose $d_T(u) = i$ with $i \geq \Delta(T) - \log n$. Then $d_{T'}(u) = i - 1$, so the decrease in $\Phi(u)$ is $3^i - 3^{i-1} = 2 \cdot 3^{i-1}$. The increase of $\Phi(v)$ is $3^{d_{T'}(v)} - 3^{d_T(v)} \leq 3^{i-1} - 3^{i-2} = 2 \cdot 3^{i-2}$, and the same is true for $\Phi(w)$.

So the overall decrease in Φ is at least

$$\begin{aligned}
2 \cdot 3^{i-1} - 4 \cdot 3^{i-2} &= \frac{2}{9} \cdot 3^i \\
&\geq \frac{2}{9} \cdot 3^{\Delta(T) - \log n} \\
&= \frac{2}{9 \cdot 3^{\log n}} 3^{\Delta(T)} \\
&\geq \frac{2}{9n^{\log 3}} 3^{\Delta(T)} \\
&\geq \frac{2}{9n^2} \cdot \frac{1}{n} \Phi(T) \\
&= \frac{2}{9n^3} \Phi(T).
\end{aligned}$$

Note that x (the other endpoint of the edge incident on u that we removed to add $\{v, w\}$) might also have a different degree in T' than in T, but in this case its degree will be smaller so this only helps us. ■

Suppose we run $\frac{9}{2} n^4 \ln 3$ iterations. Then $\Phi(T) \leq (1 - \frac{2}{9n^3})^{-\frac{9}{2} n^4 \ln 3} \cdot n 3^n \leq n$. As $\Phi(T) \geq 3n$, this means that the algorithm must stop in less than $\frac{9}{2} n^4 \ln 3$ iterations. ■

6.3.2 Approximation Ratio

A *Locally Optimal Tree* (LOT) is a tree T in which there are no u-improvements for any u with $d_T(u) \geq \Delta(T) - \log n$. In other words, a LOT is a tree which the local search algorithm might output. Let T^* denote the optimal tree, and Δ^* its maximum degree.

Theorem 6.3.4 *Let T be a LOT. Then $\Delta(T) \leq 2 \cdot \Delta^* + \log n$.*

To prove this, we're going to need a few definitions and lemmas. First, we're going to need a way to lower bound Δ^*, which is the point of the next lemma.

Lemma 6.3.5 *Consider partitioning G into pieces $V_1, V_2, \ldots, V_k$, and let $E' \subseteq E$ be the edges with endpoints in different parts of the partition. Let $V' \subseteq V$ be a vertex cover for E' (every edge in E' has at least one endpoint in V'). Then $\Delta^* \geq \frac{k-1}{|V'|}$.*

Proof: Let T^* denote the optimal tree. Clearly any spanning tree, including T^*, must have at least $k-1$ edges between the pieces. So there exists some node in V' with degree at least $\frac{k-1}{|V'|}$ in T^*, and thus $\Delta^* \geq \frac{k-1}{|V'|}$. ■

While we want to use this lemma to lower bound Δ^*, we're going to need to figure out which partition to use. Let's give some more definitions and lemmas. Note that throughout this analysis, we will only care about $i \geq \Delta^* - \log n$.

Definition 6.3.6 *Let S_i be the nodes with degree at least i in T.*

Claim 6.3.7 *There exists an $i \geq \Delta(T) - \log n$ such that $|S_{i-1}| \leq 2|S_i|$.*

Proof: Suppose this is false. Then $|S_{i-1}| > 2|S_i|$ for all $i \geq \Delta(T) - \log n$. Since $|S_{\Delta(T)}| \geq 1$, this implies that $|S_{\Delta(T)-\log n}| > n$, a contradiction. ■

This is going to be the value of i that we end up using; let i^* be the value of i from Claim 6.3.7 applied to T. For now, to make notation easier, let's prove a few facts that hold for all i (not just i^*), even though in the end we'll only apply them to i^*.

Definition 6.3.8 *Let E_i^T be edges of T incident on nodes in S_i.*

Claim 6.3.9 $|E_i^T| \geq (i-1)|S_i| + 1$

Proof: By the definition of S_i, we have that

$$\sum_{u \in S_i} d_T(u) \geq i|S_i|.$$

Now the number of edges $\{u, v\} \in T$ with $u, v \in S_i$ is at most $|S_i| - 1$ because T is a tree. This implies that

$$|E_i^T| \geq i|S_i| - (|S_i| - 1) = (i-1)|S_i| + 1$$

as claimed. ■

Now let E_i^G be edges in G between components of $T - E_i^T$.

Claim 6.3.10 *Let $e = \{x, y\} \in E_i^G$. Then either x or y in S_{i-1}.*

Proof: There are two cases to consider:

Case 1: $e \in E_i^T$. Then x or y is in $S_i \subseteq S_{i-1}$.

Case 2: $e \notin E_i^T$. Suppose that neither x nor y is in S_{i-1}. Since e goes between components of $T - E_i^T$, if we add e to T it closes a cycle which contains some edge $\{u, v\} \in E_i^T$. So either u or v is in S_i – without loss of generality we will assume that $u \in S_i$. But then $(u, \{x, y\})$ is a u-improvement, since $d_T(u) \geq i$ and by assumption both x and y have degree in T at most $i-2$. This is a contradiction, since T is a LOT. ■

Let i^* be the value from Claim 6.3.7 applied to T. Consider the partition of V defined by the components of $T - E_{i^*}^T$. There are $|E_{i^*}^T|$ components in the decomposition, and by Claim 6.3.10, the set of vertices incident on edges that cross this partition is a subset of S_{i^*-1}. Thus we can finally

apply Lemma 6.3.5 to get:

$$\begin{aligned}\Delta^* &\geq \frac{|E^T_{i^*}| - 1}{|S_{i^*-1}|} \\ &\geq \frac{(i^* - 1)|S_{i^*}|}{2|S_{i^*}|} \\ &\geq \frac{i^* - 1}{2} \\ &\geq \frac{\Delta(T) - \log n}{2}.\end{aligned}$$

This implies that $\Delta(T) \leq 2\Delta^* + \log n$.

6.4 Extensions

This algorithm and result is due to Fürer and Raghavachari [FR94]. They actually gave a much stronger version of this algorithm, which is still based on local search but is much more involved, which improves over the above result in two ways:

1. Instead of $\Delta(T) \leq 2\Delta^* + \log n$, they get the much stronger bound $\Delta(T) \leq \Delta^* + 1$. In other words, they're within an additive 1 of the optimum!
2. Their algorithm even works for the min-degree *Steiner Tree* problem, rather than min-degree spanning tree.

I'm particularly interested in this problem from a distributed and parallel point of view – if you think about it briefly, you'll realize that local search does not interact well with distributed computation, since we generally assume that we make only a single local move at a time. However, we proved a few years ago that the minimum-degree spanning tree problem is related to the "network capacity" of something called the "mobile telephone model" [DHNW19]. So we designed a distributed version of this local search algorithm, which has a weaker guarantee that $\Delta(T) \leq O(\Delta^* + \log n)$ [DHIN19]. If you're interested in trying to improve this (or related problems), come talk to me!

References

[DHIN19] Michael Dinitz, Magnús M. Halldórsson, Taisuke Izumi, and Calvin Newport. Distributed minimum degree spanning trees. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, PODC '19, pages 511–520, 2019.

[DHNW19] Michael Dinitz, Magnús M. Halldórsson, Calvin Newport, and Alex Weaver. The Capacity of Smartphone Peer-To-Peer Networks. In *33rd International Symposium on Distributed Computing (DISC 2019)*, pages 14:1–14:17, 2019.

[FR94] Martin Fürer and Balaji Raghavachari. Approximating the minimum-degree steiner tree to within one of optimal. *J. Algorithms*, 17(3):409–423, 1994.