15-854: Approximations Algorithms **Lecturer:** R. Ravi
Topic: Local Search: Min Degree Spanning Tree **Date:** 9/28/2005
Scribe: Florin Oprea

6.1 Min Degree Spanning Tree

In this lecture we continue the proof of the approximation guarantee given by local search for the min degree spanning tree problem. The proof is due to Furer and Raghavachari ([1]).

Let LOT be a locally optimal spanning tree (one in which no `T-improvment` can be made). Then adding an edge $f = (x, y) \in G \setminus T$ to the LOT will create exactly one cycle (see figure 6.1.1). By local optimality, the vertices of the cycle different from x and y will have a degree no greater than $\deg_T(x) + 1$ (assuming x has the higher degree between x and y).

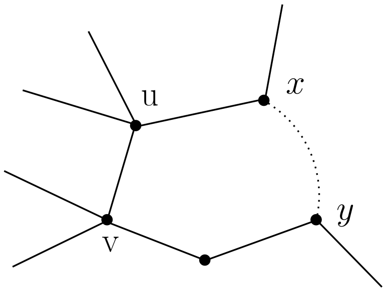

Figure 6.1.1: Condition for local optimality

Let Δ_{LOT} be the maximum degree of LOT and let Δ^* be the maximum degree of an optimal solution to the min degree spanning tree problem for G. The following theorem shows that any LOT is not far from the optimal (it is in fact within an additive log factor from a constant factor aproximation).

Theorem 6.1.1 *For any LOT,*

$$\Delta_{LOT} \leq 2\Delta^* + \log_2 n \tag{6.1.1}$$

One can replace the constant 2 by any integer $b \geq 3$.

Recall the notion of a *witness* from the previous lecture: this was just a subset $X \subseteq V$ that provided a lower bound $W(X)$ for the maximum degree of any spanning tree T. Here we use a slightly different type of witness: this will now consist of a pair (X, Π), where Π is a partition of the vertices of V. Let us call this a *modified witness*. The only requirement for Π so that (X, Π) is a modified witness is the following: all edges of a spanning tree that cross Π have to be incident to at least one node of X.

Denote by $r(\Pi)$ the *rank* of Π: this is just the number of sets in Π. A spanning tree T of G needs at least $r(\Pi)-1$ edges to connect the sets of Π between them. Also, if (X, Π) is a modified witness, each such edge is incident to some node in X. This gives us the following lemma:

Lemma 6.1.2 *For a modified witness* (X, Π)*, the following is true:*

$$\Delta^* \geq \frac{r(\Pi)-1}{|X|} \tag{6.1.2}$$

Now let us see how can we define a good witness for Δ^*. Lemma 6.1.2 suggests the following property of such a witness: the rank $r(\Pi)$ should be large, while the set X should have a small number of nodes. Thus, ideally, the set X would include only high degree nodes whose removal would create a large number of components. Let us then start with the following set X:

$$X = \{v \mid \deg_{LOT}(v) = \Delta_{LOT}\} \tag{6.1.3}$$

Note that this is actually our best witness for LOT. Unfortunately, what we need is a witness in the graph G not in LOT and the connected components formed in T after the removal of X might not be all distinct in the original graph G (there might be some edges in $G \setminus T$ connecting them). Thus we could have a large number of components in T but a very small number in G. We will now modify X such that the number of connected components in G after the removal of X is still large.

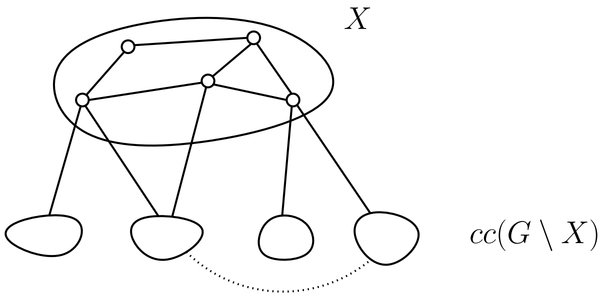

Figure 6.1.2: A modified witness for Δ^*

Consider an edge in $G \setminus T$ connecting two components of T after the removal of X (see figure 6.1.2). Such an edge has to have at least one endpoint of degree $\Delta_{LOT}-1$ since otherwise, by the observation from the beginning of the lecture, we could make an improvement. Since LOT is locally optimal, no such improvement can be made. Therefore, if we include in X all the nodes of degree $\Delta_{LOT}-1$ the connected components resulted in T after the removal of X are still disconnected in G after the removal of X. Thus, our new X will be:

$$X = \{v \mid \deg_{LOT}(v) = \Delta_{LOT}\} \cup \{v \mid \deg_{LOT}(v) = \Delta_{LOT}-1\} \tag{6.1.4}$$

To obtain the bound from theorem 6.1.1, our set X might actually need to be larger than that. The following proof clarifies how much larger X needs to be.

Proof of Theorem 6.1.1: Consider the sets $S_d = \{v | \deg_{LOT}(v) \geq d\}$. Note that $(S_d)_{1 \leq d \leq \Delta_{LOT}}$ is a decreasing sequence ($S_{d-1} \supseteq S_d$) and $S_{\Delta_{LOT}} \neq \phi$. We make the following claim:

Claim 6.1.3 *There exists a* $d \in \{\Delta_{LOT}, \Delta_{LOT} - 1, \ldots, \Delta_{LOT} - \log_2 n\}$ *such that* $|S_{d-1}| \leq 2|S_d|$.

Proof: Suppose this is not true. Then $n = |S_1| > 2|S_2| > \ldots > 2^{\log_2 n}|S_{\Delta_{LOT}}| \geq n \cdot 1 = n$, a contradiction. ■

Let d be a number satisfying the property from lemma 6.1.3. Let Π be the partition formed by the singleton sets made of elements of S_d and by the node sets of the trees remaining in LOT after the removal of S_d. We will use the pair (S_{d-1}, Π) as our modified witness for Δ^*.

To provide a lower bound for the rank of Π let us count the number of edges incident to vertices of S_d in LOT. There are $|S_d| \cdot d$ such edges due to the nodes of S_d having degree at least d, and we may have counted at most $|S_d| - 1$ of them twice (these are edges between nodes of S_d). Thus, there are at least $|S_d|(d-1)+1$ distinct edges connecting nodes in S_d between themselves and with the connected components. Since LOT is a tree, it has one more vertex than the number of edges, implying a number of at least $|S_d|(d-1)+2$ connected components and vertices in S_d. The rank of Π therefore satisfies: $r(\Pi) \geq |S_d|(d-1)+2$.

By lemma 6.1.2 and the choice of d we have:

$$\Delta^* \geq \frac{|S_d|(d-1)+2-1}{|S_{d-1}|} \geq \frac{|S_d|(d-1)+1}{2|S_d|} > \frac{1}{2}(d-1) \tag{6.1.5}$$

Thus $d < 2\Delta^* + 1$. On the other hand, by claim 6.1.3, $d > \Delta_{LOT} - \log_2 n$. It follows that $\Delta_{LOT} < 2\Delta^* + \log_2 n + 1$ which is just another form of writing 6.1.1. ■

This completes the proof of the approximation guarantee of a solution obtained through local search. However, this does not tell us how long it will take before the algorithm completes. It might be the case that an exponential number of steps might be needed to reach such a LOT. To provide a polynomial time algorithm, we will next consider POTs: a POT (pseudo-local optimal tree) is just a tree that permits no local improvement involving any node of degree $\Delta - \log_2 n$ or higher.

The previous theorem is also true for POTs. We will now make a potential function argument that will show that a POT can be found in polynomial time.

Lemma 6.1.4 *Starting from any tree T we can arrive at a POT in polynomial time.*

Proof: Consider the following potential function:

$$\Phi_v = 3^{\deg_T(v)} \tag{6.1.6}$$

$$\Phi_T = \sum_{v \in V} \Phi_v \tag{6.1.7}$$

Note that $\Phi_T \leq n \cdot 3^{\Delta_{LOT}}$. Let us now compute the change in potential ($\Delta\Phi$) when we perform a local improvement involving a vertex of degree $d \geq \Delta_{LOT} - \log_2 n$. Assume edge xy is added to the tree and edge uv is removed from the tree. Let $d_v = \deg_T(v)$ be the degree of v in T. Assume wlog that $d_x \geq d_y$. We consider only the terms in Φ_T that change during the improvement:

$$\Phi^{\text{init}} = 3^{d_v} + 3^{d_u} + 3^{d_x} + 3^{d_y} \tag{6.1.8}$$

$$\Phi^{\text{final}} = 3^{d_v-1} + 3^{d_u-1} + 3^{d_x+1} + 3^{d_y+1} \tag{6.1.9}$$

$$\Delta\Phi = 3^{d_v-1}(3-1) + 3^{d_u-1}(3-1) - 3^{d_x}(3-1) - 3^{d_y}(3-1) \tag{6.1.10}$$

Since what we consider is an improvement, we have that both d_x+1 and d_y+1 are less than d_u-1 (say $d_u \geq d_v$). Therefore the sum of the last three terms of $\Delta\Phi$ is at least $\frac{2}{3}3^{d_u-1}$. We then have:

$$\Delta\Phi \geq \frac{2}{3} \cdot 3^{d_u-1} \geq \frac{2}{3} \cdot \frac{3^{\Delta_{LOT}}}{3^{\log n}} \geq \frac{2}{3} \cdot \frac{\Phi_T}{3^{\log n} n} \geq \frac{\Phi_T}{\Omega(n^2)} \tag{6.1.11}$$

This implies that after $O(n^2)$ improvements Φ_T decreases by a constant. Hence after $O(n^3)$ improvements we have arrived at a POT. Since each improvement can be executed in polynomial time, it follows that we can reach a POT in polynomial time. ■

References

[1] Martin Furer and Balaji Raghavachari. Approximating the minimum degree spanning tree to within one from the optimal degree. In *SODA '92: Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, pages 317–324, 1992.