1 Introduction

This lecture starts by reviewing the main points and differences between the local ratio algorithms studied previously. We then present the NP-Hard problem Steiner tree and show a 2-approximation algorithm for it. A different approximation algorithm is following, this time using the local ratio technique. This algorithm generalizes naturally to the Generalized Steiner Forest problem, introduced at the end of this lecture.

2 VC and IS Local Ratio Algorithms Recap

<table>
<thead>
<tr>
<th>Algorithm Step</th>
<th>ISB-2</th>
<th>ISD-2</th>
<th>IS-1</th>
<th>VC-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throw useless input. If the input is trivial, return a solution with the required approximation</td>
<td>$v_i \leq 0$</td>
<td>$v_i \leq 0$</td>
<td>$v_i \leq 0$</td>
<td>$-$</td>
</tr>
<tr>
<td>Break weights into degenerated set and general (smaller) set</td>
<td>Take the first ending interval</td>
<td>Take the first ending copy. All intersecting intervals will be of value $v$</td>
<td>Take the first ending interval. All intersecting intervals will have value $v$</td>
<td>Return the less costly vertex incidenting the edge</td>
</tr>
<tr>
<td>Recursive solution</td>
<td>First interval removed</td>
<td>First living copy removed</td>
<td>First interval removed</td>
<td>There’s at least one more zero-valued vertex.</td>
</tr>
<tr>
<td>Fix the solution to match the original problem, keeping the approximation ratio</td>
<td>Add interval if possible (total length of intersecting intervals $\leq 1 - w$)</td>
<td>Add interval copy if possible (if no other interval intersects and no other copy was taken)</td>
<td>Add the first interval if possible (recall its value is 0)</td>
<td>No fixing is required. Accepted solution is the recursive solution</td>
</tr>
<tr>
<td>Trivial input</td>
<td>$\phi \rightarrow \phi$</td>
<td>$\phi \rightarrow \phi$</td>
<td>$\phi \rightarrow 0$</td>
<td>zero-valued vertices form VC</td>
</tr>
<tr>
<td>Proof: $OPT \geq T$, ANY $\leq T\alpha$ ($\alpha$ is the approximation ratio)</td>
<td>$ANYFIXED \geq B, OPT \leq 2B$</td>
<td>$ANYFIXED \leq 2V$, $OPT \geq V$</td>
<td>$OPT \geq L$, $ANYFIXED = V$</td>
<td>$ANYFIXED \leq 2V_1$, $OPT \geq V_1$</td>
</tr>
</tbody>
</table>

**Theorem.** If there exists a solution $S$ such that

$$w_1(S) \leq \alpha OPT(I, w_1)$$

$$w_2(S) \leq \alpha OPT(I, w_2)$$

Then

$$w(S) \leq \alpha OPT(I, w_1 + w_2).$$
3 Steiner Tree

Definition. Let $G = (V, E)$ a connected, edge-weighted graph and $S \subseteq V$. A Steiner Tree with respect to $G, S$ is a minimum-weight subgraph of $G$ spanning all the vertices in $S$.

3.1 2-Approximation Algorithm

- Reconstruct the edges of $G$, such that the distance between $u, v \in S$ is the weight of the shortest path connecting them. Denote the subgraph of $G$ induced by the vertices of $S$ by $G'$.
- Find $MST$ $T$ on $G'$.
- Translate $T$ back into a solution to the original graph $G$ by converting edges back into paths and throwing spare edges.

Claim 1. The algorithm above is a 2-approximation algorithm.

Note. Any spanning tree for $G'$ is a feasible solution for the Steiner Tree problem.

Proof of Claim 1. Consider a Steiner tree of cost $OPT$. Double the edges of $OPT$ to obtain an Euler graph. Thus there is a cycle (traverses once in each edge) of weight $2OPT$. Shortcutting this cycle (by skipping vertices that were already visited or not belonging to $S$) will yield a new cycle in $G'$ comprised of all vertices of $S$. Clearly, this operation does not increase the cost of the solution, as the shortcutting edges represent the weight of the shortest path between each pair of vertices in $S$. Finally we have a spanning tree of $G'$ of cost $2OPT$ at most. This is an upper bound for the $MST$ $T$ of $G'$ returned by our algorithm. Therefore a 2-approximation for the problem is achieved.

Note. There exist algorithms with a better approximation ratio in the general case, and particularly in an Euclidean graph.

3.2 Approximating Steiner Tree by Local Ratio Algorithm

Let $G = (V, E), S \subseteq V$. The following algorithm uses the local ratio technique to approximate Steiner tree problem by ratio 2.

Denote:

- $\varepsilon_1 = \min\{c(u, v) | u \in S \text{ xor } v \in S\}$
- $\varepsilon_2 = \min\{\frac{1}{2}c(u, v) | u \in S \text{ and } v \in S\}$
- $\varepsilon = \min\{\varepsilon_1, \varepsilon_2\}$

Algorithm Steiner-Tree.

1. if there exists $e \in E$ such that $c(e) = 0$ then contract $e$ (identify the two vertices inciding in $e$. The edges of the new vertex will be the union of the edges of the contracted vertices. In case of double edges, use the cheaper one for the new vertex).
   if $|S| = 1$ return the vertex in $S$ and no edges.

2. for each $e = (u, v) \in S$, the value of $e$ would be $\varepsilon$ if $|\{u, v\} \cap S| = 1$ or $2\varepsilon$ if $|\{u, v\} \cap S| = 2$. All other edges will have zero value.
   Perform the contraction recursively, treating the new vertex as a vertex in $S$ if any of its contracted vertices were in $S$. 
3. Solve by recursion. Finally, expand the contracted vertices and add the edge connecting them with zero weight.

4. Leave only necessary edges in the graph (that is, all the leaves should be vertices in S).

**Claim 2.** Any fixed solution with degenerated weights is a 2-approximation to $OPT$, namely

- $OPT \geq |S| \cdot \varepsilon$
- $ANYFIXED \leq 2\varepsilon(|S| - 1)$

**Proof of Claim 2.** We start by showing that $OPT \geq |S| \cdot \varepsilon$. By induction on $|S|$. If $|S| = 2$ the minimal cost of the spanning tree would be $2\varepsilon$, whether the vertices are connected directly or not. For a larger $S$, we have

$$OPT_S \geq OPT_{S-1} + \varepsilon \geq (|S| - 1) \cdot \varepsilon + \varepsilon = |S| \cdot \varepsilon.$$ 

This is due to the fact that the minimal cost of adding a vertex to the set $S$ is $\varepsilon$. The second inequality is due to the induction hypothesis.

We now show that $ANYFIXED \leq 2\varepsilon(|S| - 1)$. By induction on $|S|$. If $|S| = 2$, the fixed solution is exactly $2\varepsilon = 2\varepsilon(2 - 1)$. For larger $S$, consider the given solution. Traverse from a vertex in $S$ until reaching another vertex in $S$ or a vertex with degree at least 2. Fold the created path to remain with a tree containing $|S| - 1$ leaves in $S$. By induction,

$$A(|S|) \leq A(|S| - 1) + 2\varepsilon \leq 2(|S| - 2) \cdot \varepsilon + 2\varepsilon = s(|S| - 1) \cdot \varepsilon.$$ 

### 4 Generalized Steiner Forest

Let $G = (V, E)$ undirected weighted graph, $\{(s_i, t_i)\}_{i=1}^k$ pairs of vertices in $G$. A feasible solution to the problem would be a subgraph of $G$ (possibly unconnected) connecting $s_i$ and $t_i$ for all $i \in [k]$. The objective is to find a feasible subgraph which is **minimal**.

Two immediate solutions can be easy proved to be very costly:

- Take all the shortest paths $s_i$ to $t_i$. This can be as bad as $|S| \cdot OPT$.
- Take the Steiner tree over all the vertices $s_i, t_i$. This can be unlimited worse than $OPT$.

**Note.** Steiner tree is a special case of this problem by taking all pairs of vertices in $|S|$ (there are $\binom{|S|}{2}$ such pairs).

**Note.** An equivalent presentation of the problem is as follows: Given $n$ sets of vertices $S_1, S_2, ..., S_n$, find a minimal forest such that for all $u, v \in S_i$ there exists a path from $u$ to $v$ in the forest.

**Proof of the Note.** Consider the alternative presentation. For all $i$ and for each $u, v \in S_i$, construct a pair $(u, v)$. This gives an equivalent problem with the original presentation. For the other direction, simply construct a set of two vertices for each pair of the original presentation.