CS602 Applied Algorithms **2019-20 Sem II**

Lecture 14: March 2

Scribe: Rohit Kumar Singh *Lecturer: Rohit Gurjar*

1 Primal Dual for Approximation Algorithm

The following two algorithms are can be viewed as primal-dual algorithms in disguise:

- Dijkstra's algorithm
- Kruskal's algorithm

2 Soap Bubbles

2.1 Experiment

If two glass plates/thermacol with verticle toothpicks between them are dipped into soapy water, then a soap film will form connecting all the toothpicks. Most of the times, this film is such that it minimizes total surface area, which corresponds to minimum energy configuration, while connecting all toothpicks.

2.2 Result

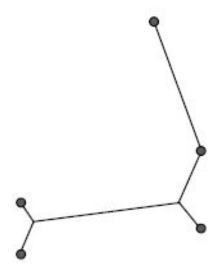

Figure 1: A Steiner tree connecting points.

Given a set of points in the Euclidean plane, a (Euclidean) Steiner tree (Figure 1) is a collection of line segments of minimum total length connecting the points, where the segments can meet at vertices (called Steiner vertices) other than the given points themselves.

For $N = 3$ points, there are two possible cases:

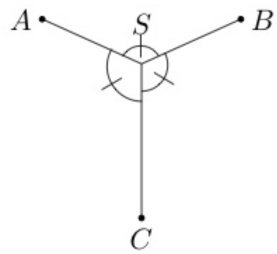

Figure 2: Steiner tree for 3 terminals [5]

(a) If the triangle formed by the given points has all angles less than 120°, then the solution is given by a Steiner point located in the middle of the triangle.

(b) When one of the angles is more than 120°then the solution is given by the two sides of the triangle which meet this angle.

For $N = 4$ points, there are two solutions, as shown here. One is just a rotation of the other by 90°.

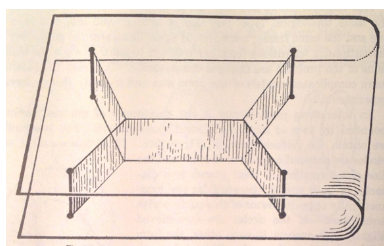

Figure 3: Steiner tree for 4 terminal [1]

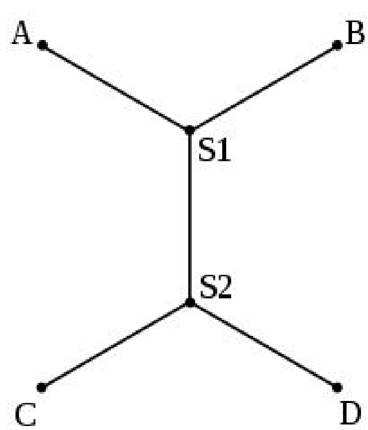

Figure 4: Alternate Steiner tree for 4 terminal [2]

For a unit square, each of these networks is of length $1 + \sqrt{3}$. A film forms comprised of plane sections meeting at angles of 120°. Soap films provide a local minimum configuration, but not always the global minimum. For $N = 5$ points, the following figure shows an example of a Steiner tree.

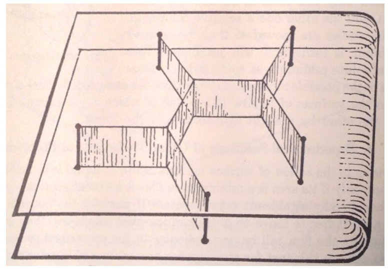

Figure 5: Steiner tree for 5 terminal [1]

To find an Euclidean Steiner tree is known to be NP-hard. This raises the following philosophical question: can NP-hard problems be solved in polynomial time using the resources of the physical universe, which is soap water in this case?

3 Steiner Tree on graph

Given an undirected graph with non-negative edge costs and a subset of vertices, usually referred to as terminals T $\subseteq V$, the Steiner tree problem in graphs requires a tree of minimum cost that contains all terminals (but may include additional vertices). Steiner tree problem is NP-hard.

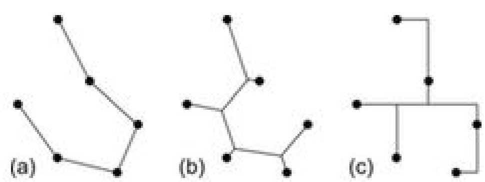

Figure 6: a) represents a minimum spanning tree, b) is an Euclidian Steiner tree and c) is a Rectilinear Steiner tree. [4]

As opposed to minimum spanning trees (figure 6: a) where all paths go through the vertices themselves, Steiner trees (figures 6: b and 6: c) allow you to have extra mediating vertices (Steiner points) resulting in a shorter cumulative cost/length. This, however, also makes the problem hard since it is difficult to find out which extra mediating point should be there.

There is more general version of this problem called Steiner forest, where we are given a collection of subsets of vertices and the goal is ensure that each subset in the collection is connected within (using additional vertices).

4 Linear Programming Formulation via hitting-sets

The first step towards designing an approximation algorithm is to express the Steiner tree/forest problem as an integer linear program. To do that we will first write a hitting set formulation.

Hitting Set: Let E be a set with each element having a cost and $S_1, S_2, .., S_m \subseteq E$ be its subsets. The hitting set problem asks for the minimum cost subset $X \subseteq E$ such that X intersects S_i for each $1 \leq i \leq m$. It can be expressed as the following linear program.

$$\begin{array}{rcl} \text{min} & \sum_{e \in E} x_e & \text{s.t.} \\ \text{for each } e \in E & x_e \geq 0 & \\ \text{for each } 1 \leq i \leq m & \sum_{e \in S_i} x_e \geq 1 & \end{array}$$

- The vertex cover problem is a special instance of hitting set problem: think of each edge as a subset of (two) vertices.
- Moreover, the set cover problem is equivalent to the hitting-set problem.

This can be seen by observing that an instance of set covering can be viewed as an arbitrary bipartite graph, with sets represented by vertices on the left, the universe represented by vertices on the right, and edges representing the inclusion of elements in sets. The task is then to find a minimum cost subset of left-vertices which covers all of the right-vertices. In the hitting set problem, the objective is to cover the left-vertices using a minimum subset of the right vertices.

- Thus, the edge cover problem also reduces to hitting set problem.

Homework Question. Can we express the Steiner tree problem as a hitting set problem?

5 Extra Reading Materials & References

1. `https://thatsmaths.com/2015/01/29/the-steiner-minimal-tree`
2. `https://en.wikipedia.org/wiki/Steiner_tree_problem`
3. `http://theory.cs.uni-bonn.de/info5/steinerkompendium/netcompendium.pdf`
4. `https://www.science.ku.dk/english/press/news/2014/the-steiner-tree-research-at-diku-experiences-a-revival`
5. `http://images.math.cnrs.fr/`