Lecture 5: Local Search

Lecturer: Allan Borodin *Scribe: Lily Li*

5.1 Continuing from Last Time...

Maximum independent set in $k+1$ claw free graphs using local search. Apparently we do better if the graph is undirected. And there are two variants of the local search: oblivious and non-oblivious (dependent on a potential function). One possible potential function is: small heavy edges are preferable to light edges (you need to pick fewer vertices to get the same weight). Thus his potential function is just the sum of the square of the weights obtained.

5.2 Max Flow

5.2.1 Ford Fulkerson

A flow network $\mathcal{F} = (G, s, t, c)$ consisting of a **bi-directional** graph $G = (V, E)$, s is source, t is terminal and c is non-neg real valued capacity. We require: (1) capacity constraint ($f(e) \leq c(e)$), (2) skew symmetry ($f(u, v) = -f(v, u)$), and (3) flow conservation (for all internal node, flow-in equals flow-out).

The goal is to find the most flow from s to t. It uses augmenting paths and is an example of local search (as was previously defined). Basically you always take the s-t path with greatest flow (each path is constrained by the edge with least capacity). Update the graph to form a new residual graph. Repeat as long as a there are such s-t paths. Do you still remember the max-flow min-cut theorem?

There could be exponentially number of augmenting paths if you are doing things funny. Edmonds and Karp proved the first polynomial algorithm by taking the shortest augmenting path (running time: $O(|V||E|^2)$. Dinitz had something with leveled graphs and blocking flows. Comes out to $O(|V|^2|E|)$.

The above regards only unweighted edges. This type of graph can be used to solved the bipartite matching problem.

5.2.2 Metric Labeling Problem

Given graph $G = (V, E)$, a set of labels $L = \{a_1, ..., a_r\}$ in a metric space M with distance γ and cost function κ. The goal is to construct an assignment α of the labels to the nodes to minimize some cost of mis-labeling (want close vertices to have similar labels, but can't just label everything the same).

5.2.3 Application of Min-cut Max-flow

Arkin-Silverberrg reduction of Weighted Interval Scheduling Problem (WISP) to a min-cost flow problem. Achieves $O(n \log n)$ (independent of the number of machines m, unlike DP techniques).

5.3 Linear Programming

... most often solved by some variant of the simplex method. You will recall that IP is an NP-Hard problem. There are polynomial time algorithm for LP using the ellipsoid and interior points method.

5.3.1 Weighted Vertex Cover

We can formulated the weighted vertex cover problem as an IP. Then we relax our problem to an LP and solve. Finally we take our LP solution (where $0 \leq x_i \leq 1$) for each variable and round to a valid IP solution. This results in a 2-approximation for the optimum IP solution.