CMPUT 675: Approximation Algorithms	**Winter 2005**

Lecture 12: Feb 18

Lecturer: Mohammad R. Salavatipour *Scribe: Zhuang Guo*

Recall from the last lecture that the minimum multicut problem is NP-hard even if the graph G is restricted to trees that are stars and we gave a factor 2 approximation algorithm for when G is a tree. In this lecture, we discuss an $O(\log k)$-factor approximation algorithm for the minimum multicut problem on general graphs.

12.1 The Minimum Multicut Problem (general graphs)

Recall the definition of multicut:
Input: An undirected graph $G = (V, E)$ with nonnegative weight/capacity c_e for each edge $e \in E$ and a set of source-sink pairs $S = \{(s_1, t_1), (s_2, t_2), \cdots, (s_k, t_k)\}$, where each pair is distinct.

Goal: Find a multicut with minimum weight.

For each vertex pair (s_i, t_i), let P_i denote the set of all paths from s_i to t_i. Let $P = \bigcup_{i=1}^{k} P_i$. Consider the LP-formulation of the minimum multicut problem:

$$\begin{array}{llr} \textit{minimize} & \sum_{e \in E} c_e \cdot x_e & \\ \textit{subject to} & \sum_{e \in p_i} x_e \geq 1, & p_i \in P_i, 1 \leq i \leq k \\ & x_e \geq 0, & e \in E \end{array}$$

The above LP-formulation has an exponential number of constraints. However, we can still solve this LP using the Ellipsoid method. Given a (possible) solution vector $\vec{x}$ (assignments to $x_e, e \in E$), we can check if it is feasible in polynomial time (this implies that the separation oracle for this LP is in P). To do so we interpret variable x_e as a distance label on edge e, for each $e \in E$; then we compute the lengths of shortest paths between each source-sink pair (s_i, t_i) w.r.t the current distance labels. If all the lengths are ≥ 1, then all the paths between each pair (s_i, t_i) must have lengths ≥ 1, and therefore, we can conclude that all constraints are satisfied and the solution is feasible. If the shortest path is < 1 then we obtain a violated constraint.

12.2 The $O(\log k)$-factor approximation Algorithm

Now, we introduce the beautiful $O(\log k)$-factor approximation algorithm due to Garg, Vazirani, and Yannakakis [GVY]. Before giving the algorithm, we restate the problem as a pipe system. This will help to some intuition behind the algorithm.

Consider a feasible solution $\vec{x}$ to the LP. Suppose that we have a pipe running between i, j if there is an edge $e = (i, j)$ in E. Let the length of this pipe be x_e and the cross-sectional area of this pipe be c_e. Therefore, $c_e \cdot x_e$ will be the volume of this pipe and $\sum_{e \in E} c_e \cdot x_e$ will be the total volume of the pipes in our system. With this definition, the multicut problem is in fact the question of designing a pipe system such that the distance between every source-sink pair is at least 1 and the total volume of pipes in the system is minimized. Therefore, the fractional optimal solution, i.e. the solution to the LP, is the volume of the pipe system and we denote it by V^*.

Definition 12.1 *For a feasible solution $\vec{x}$, denote $d_x(u,v)$ to be the length of the shortest path between u and v in G w.r.t the distance labels of $\vec{x}$.*

Definition 12.2 *For a set of vertices $S \subseteq V$, denote the set of edges in the cut $(S, V-S)$ as $\delta(S)$.*

Definition 12.3 *For a vertex $v \in V$ and a (real) radius r, define the set of vertices in G with distance $\leq r$ (with respect to distance label given by $\vec{x}$) to v as $B_x(v,r)$, i.e. $B_x(v,r) = \{u | d_x(u,v) \leq r\}$.*

The algorithm will find disjoint sets of vertices $S_1, \ldots, S_{\ell \leq k}$, called regions by growing balls around terminals such that:

- no region contains any source-sink pair, and for each $1 \leq i \leq k$, either s_i or t_i is in one of the S_j's.
- For each region, the weight of $\delta(S_j)$ is "small".

GVY $O(\log k)$-approximation algorithm

> $C \leftarrow \Phi$.
> let $\vec{x}$ be an optimal solution to the LP.
> while there is some pair (s_i, t_i) not separated, do:
>   let $S = B_x(s_i, r)$ for some $r < \frac{1}{2}$.
> $C \leftarrow C \bigcup \delta(S)$.
> $V \leftarrow V - S$.
> return C.

Lemma 12.4 *The algorithm terminates.*

Proof: Since the ball grown around a terminal s_i at each iteration has radius at most $\frac{1}{2}$ it cannot contain t_i. Thus, $\delta(S)$ will separate (at least) one source-sink pair. The algorithm has at most k iterations. ■

Lemma 12.5 *The algorithm returns a multicut.*

Proof: The only problem is when the algorithm separates some pair (s_i, t_i) and there is a pair (s_j, t_j), such that both $s_j \in B_X(s_i, r)$ and $t_j \in B_X(s_i, r)$. In this case, since we are removing all the vertices of the ball around s_i, then (s_j, t_j) will not be separated by the algorithm. But this scenario is impossible to happen. Otherwise, from $d_x(s_j, t_j) \leq d_x(s_i, s_j) + d_x(s_i, t_j) \leq 2 \cdot r < 1$, one of the LP constraints for s_j, t_j is violated. ■

Definition 12.6 *Let V^* be the optimal fractional solution to the LP. Given a vertex $v \in V$ and a radius r, a ball with radius r is defined. Define the volume of this ball (region) as*

$$V_x(v,r) = \sum_{e=(u,v) \in B_x(v,r)} c_e \cdot x_e + \sum_{\substack{e=(u,v) \in \delta(B_x(v,r)) \\ v \in B_x(v,r)}} c_e(r - d_x(u,v)) + \frac{V^*}{k}$$

and the cut volume of this region as

$$C_x(v,r) = \sum_{e \in \delta(B_x(v,r))} c_e$$

Note that $V_x(v,r)$ is an increasing function of r. It is a piece-wise linear function with possible discontinuities at values of r where new vertices are added to the region. Therefore, $V_X(v,r)$ is differentiable everywhere except those possible discontinuous points and

$$\frac{\mathbf{d}\, V_x(v,r)}{\mathbf{d}\, r} = C_x(v,r) \tag{12.1}$$

Lemma 12.7 *There is some $r < \frac{1}{2}$, such that $\frac{C_x(s_i,r)}{V_X(s_i,r)} \le 2 \cdot \ln(k+1)$ and we can find such an r in polynomial time.*

Proof: By contradiction, assume throughout the region growing process, starting with $r = 0$ ending at $r = \frac{1}{2}$:

$$\frac{C_x(s_i,r)}{V_x(s_i,r)} > 2 \cdot \ln(k+1).$$

This implies that

$$\frac{\mathbf{d}\, V_x(v,r)}{\mathbf{d}\, r} \cdot \frac{1}{V_x(s_i,r)} > 2 \cdot \ln(k+1).$$

Let $r_1 = 0 \le r_2 \le \cdots \le r_q = \frac{1}{2}$ be the radii at which new vertices are added to the region (s_i, r_q). For all r in (r_j, r_{j+1}):

$$\begin{array}{rcl}
\displaystyle\int_{rj}^{r_{j+1}} \frac{\mathbf{d}\, V_x(v,r)}{\mathbf{d}\, r} \cdot \frac{1}{V_x(s_i,r)} & > & \displaystyle\int_{rj}^{r_{j+1}} 2 \cdot \ln(k+1) \mathbf{d}\, r \\
& \Downarrow & \\
\ln(V_x(s_i,r_{j+1})) - \ln(V_x(s_i,r_j)) & > & (r_{j+1} - r_j) \cdot 2 \cdot \ln(k+1).
\end{array}$$

We are going to sum up over all intervals (r_j, r_{j+1}) for $1 \le j < q$. This will give us a telescopic sum and the terms will be canceled out except the first and the last term. Doing this, even though the function is discontinuous at the end-points of the intervals, is valid because the function $V_x(s_i, r)$ is an increasing function. Thus:

$$\begin{array}{rcl}
\ln(V_X(s_i,r_q)) - \ln(V_x(s_i,r_1)) & > & 2 \cdot \ln(k+1) \cdot (r_q - r_1) \\
& \Downarrow & \\
\ln(V_x(s_i,\frac{1}{2})) & > & 2 \cdot \ln(k+1) \cdot r_q + \ln(\frac{V^*}{k}) \\
& \Downarrow & \\
\ln(V_X(s_i,\frac{1}{2})) & = & \ln(k+1) + \ln(\frac{V^*}{k}) \\
& \Downarrow & \\
\ln(V_x(s_i,\frac{1}{2})) & > & \ln(V^* + \frac{V^*}{k}) \\
& \Downarrow & \\
V_x(s_i,\frac{1}{2}) & > & V^* + \frac{V^*}{k}.
\end{array}$$

But this cannot happen, since $V_x(s_i, \frac{1}{2})$ is part of the total volume and cannot be larger than it. This implies that there exists such an $r < \frac{1}{2}$. To find r, consider the vertices of G according to non-decreasing order of distance from s_i: $s_i = v_1, v_2, \ldots, v_p$ with distances $r_1 = 0 \leq r_2 \leq \ldots, r_p \leq r_{p+1}$ where $r_{p+1} \geq \frac{1}{2}$ and $r_p < \frac{1}{2}$. At any interval (r_j, r_{j+1}), the volume $V_x(s_i, r)$ increases while the value of cut $C_x(s_i, r)$ is fixed. Therefore, the volume is maximized (i.e. $\frac{C_x(s_i,r)}{V_x(s_i,r)}$ is minimized) at the end of the interval. So it is enough to check the ratio $\frac{C_x(s_i,r)}{V_x(s_i,r)}$ at the end of the intervals. ■

Theorem 12.8 *The GVY algorithm is a* $(4\ln(k+1))$*-factor approximation algorithm for IMC.*

Proof: We charge the cost of the edges removed from the graph at each iteration against the volume of the region removed. By lemma 12.7, at each iteration:

$$\begin{aligned}
C_x(s_i, r) &\leq 2\ln(k+1) \cdot V_x(s_i, r). \\
&\Downarrow \quad (\text{ Summing up for } 1 \leq i \leq k) \\
\sum_{e \in C} c_e &\leq 2\ln(k+1) \sum_{i=1}^{k} V_x(s_i, r) \\
&\leq 2\ln(k+1) \cdot (V^* + \frac{V^*}{k} \cdot k) \\
&= 4\ln(k+1) \cdot V^* \\
&\leq 4\ln(k+1) \cdot OPT.
\end{aligned}$$

■

References

GVY N. Garg, V.V. Vazirani, and M. Yannakakis, Approximate max-flow min-(multi)cut theorems and their applications, *SIAM Journal on Computing*, 1996, 25:235–251.