Network Design and Game Theory
Spring 2008
Lecture 11

Instructor: Mohammad T. Hajiaghayi
Scribe: Gayatree Ganu

April 28, 2008

1 Overview

In this lecture, we discuss the problem of OBLIVIOUS ROUTING. In oblivious routing, a system of optional paths is chosen in advance for every source-destination pair (oblivious to the demands on that pair and also oblivious to other source-destination pairs), and every packet for that pair must travel along one of these optional paths.

2 Definitions

OBLIVIOUS ROUTING

INPUT : A grapg $G(V, E)$ with capacity $C(u, v)$ on every edge (u, v), and souce-target pairs (s_i, t_i).

GOAL: Find a fixed rule for routng between all source-destination pairs which has a close to optimum congestion for any set of demands, i.e.

$$\text{COMP(OBL)} = \textbf{minimize:} \quad \max_{\text{demand-matrix D}} \left[\frac{\text{congestion}_{\text{OBL}}(D)}{\text{congestion}_{\text{OPT}}(D)} \right] \tag{1}$$

where congestion is defined as
$congestion = max_e(load(e)/capacity(e) = max_e(relativeload * load(e))$
and $load(e)$ is the sum of flow through an edge e.

3 Some Results

(1) For undirected graphs there exists Oblivious Routing Algorithm OBL, where $\mathsf{COMP(OBL)}$ is
$\mathsf{O}(\lg^3 n)$ - [Räcke '02]
$\mathsf{O}(\lg^2 n \cdot \lg\lg n)$ - [HHR '03]
$\Theta(\lg n)$ - [Räcke '08]

(2) For directed or even node weighted graphs, there exist directed graphs for which every Oblivious Routing Algorithm OBL, where $\mathsf{COMP(OBL)}$ is $\Omega(\sqrt{n})$ - [ACFKR, STOC '03] and [HKRL, SODA '05].

(3) Though the ratios for directed graphs are $\Omega(\sqrt{n})$ and $\Omega(\lg n)$ for general undirected graphs, there is a polynomial time construction that gives the rue optimal ratio for any (directed/undirected) network - [ACFKR, '03].

(4) If demands are random from known distributions, the competitive ratio is in $\mathsf{O}(\lg^2 n)$ with high probability in directed graphs.

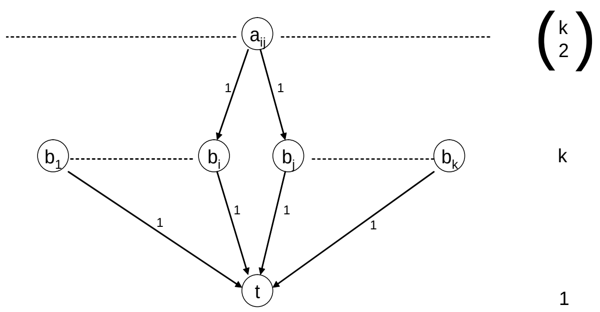
(5) Randomness does not help in undirected graphs and we have a lower bound of $\mathsf{O}(\frac{\lg n}{\lg\lg n})$ in this case.

(6) Other goal such as maximizing throughput instead of minimizing congestion has also been considered - [AHKL, SODA '05].

(7) Oblivious network design with general edge functions - [GHR, SODA '06].

4 For directed graphs $\mathsf{COMP(OBL)} = \Omega(\sqrt{n})$

Proof (by construction):
Consider the following edge-weighted directed graph.



In the above network, the commodity pairs are (a_{ij}, t). The total flow in the network is $\binom{k}{2}$.

Using the averaging argument, there is atleast one node b_x which receives $\binom{k}{2}/k$ units of flow. Hence, $\mathsf{Load}(b_x) \geq \binom{k}{2}/k = \frac{k(k-1)/2}{k} = \frac{k-1}{2}$.

Congestion (OBL) $= \frac{k-1}{2}$
However, optimum can route this demand with congestion 1 by using the paths $a_{ix} - b_i - t$ for demands from nodes a_{ix} and the paths $a_{xj} - b_j - t$ for demands from nodes a_{xj}.
Congestion (OPT) = 1

Therefore,
$\mathsf{COMP(OBL)} = \frac{k-1}{2} = \Omega(k) = \Omega(\sqrt{n})$

(**Remark:** For a node-weighted directed graph the proof is similar to the edge-weighted case proved above. Considering weight on edge (b_x, t) is same as considering weight on node b_x.)

5 Tree decomposition for $\Theta(\lg n)$ bound for undirected graphs

Räcke [FOCS '02] introduced a tree decomposition that aims at constructing a tree that does not approximate point-to-point distances in the input graph (like Bartal or FRT) but instead approximates the cut structure of the graph.

The model is as follows:
Given a graph $G(V, E)$ with nodes $|V| = n$.
Also there is a capacity or weight function on the edges of the graph.
$C(u, v) > 0$ if there is an edge from u to v in G. $C(u, v) = 0$ if $(u, v) \notin E$.
Note that G is undirected, $C(u, v) = C(v, u)$.

Decomposition Trees: A decomposition tree for the grpah G is a rooted tree $T = (V_t, E_t)$ whose leaf nodes correspond to nodes in G. There is an embedding of T into G using the following two functions:
$m_V : V_t \to V$ which is a node mapping function that maps tree nodes to nodes in the graph.
$m_E : E_t \to E^*$ which maps an edge $e_t(u_t, v_t)$ of T to a path P_{uv} between the corresponding end points in G.

We also introduce the functions $m'_V : V \to V_t$ and $m'_E : E^* \to E_t$ responsible for mapping from G to T. m'_V outputs for a node in G, the corresponding leaf node in T. m'_E outputs for an edge in G, the unique shortest path in T between the corresponding nodes.

Additionally, for a multicommodity flow f_T on the decomposition tree, we use $m(f_T)$ to denote the multicommodity flow obtained by mapping f_T to G via the edge mapping function m_E. similarly, for a flow f in G, we define $m'(f)$ as the flow in T.

Given a decomposition tree T for G we define the capacity $C(u_t, v_t)$ of a tree edge $e_t = (u_t, v_t)$ as $C(u_t, v_t) = \sum_{u \in V_{ut}, v \in V_{vt}} C(u, v)$ where V_{ut} and V_{vt} define two partitions of V induced by the cut corresponding to the edge e_T in the graph G.

Theorem 1: suppose that you are given a multicommodity flow f in G with congestion C_G, then the flow $m'(f)$ obtained by mapping f to some decomposition tree T results in a flow in T with congestion $C_T \leq C_G$.
Proof: Suppose an edge $e_t = (u_t, v_t)$ in the tree has congestion C_T. the total capacity of all traffic that traverses the cut in G between V_{ut} and V_{vt} is exactly $C(e_t)$, the capacity of the corresponding edge in T. Hence, by the averaging argument, one of these edges must have relative load of atleast C_T. Hence, $C_T \leq C_G$.

(**Remark:** For the derivation of $\Theta(\lg n)$ bound refer to professor's handwritten notes.)