Multicommodity flow

Luke Snyder and Sang-Ho Shim

1 Introduction

Leighton and Rao use multicommodity flow results to design the first polynomial time approximation algorithms for well known NP-Hard optimization problems. Such problems include graph partitioning, crossing number, VLSI layout, and many more. Furthermore, Leighton and Rao are responsible for establishing the max-flow min-cut theorems on multicommodity flow problems, which lead to the algorithms mentioned above. In this paper we will establish the definitions and lemmas necessary to understand multicommodity flow problems, and we will also present the influential max-flow min-cut theorem by Leighton and Rao.

1.1 Single commodity flow problem

The multicommodity flow problem is an extension of the single commodity flow problem. For this reason we will first formalize the definitions for single commodity flow, min-cut, and max flow, which will lead to a better understanding of the definitions under the multicommodity setting. Then we will prove that the min-cut of a 1-commodity flow problem is an upper bound for the max-flow. We will also give an overview of the Ford Fulkerson Algorithm, which turned out to be a motivating factor for the work done by Leighton and Rao.

Definition 1 A *single commodity flow problem* is a network that can be thought of as a 5-tuple (V, E, C, s, t), where

(1) V is a set of nodes

(2) E is a set of edges

(3) $C : E \to R_{\geq 0}$ is the capacity function for each edge

(4) s is the designated source node

(5) t is the designated sink node

The objective is to route as much flow as possible from the source to the sink. This must be done without violating any edge capacities and maintaining that the flow into any node equals the flow out of that node, excluding the source and the sink.

Definition 2 The *max-flow* is the maximum amount of flow that can be routed.

Definition 3 The *min-cut* is the minimum amount of capacity that needs to be removed from the network in order to disconnect the source from the sink. More formally the min-cut is

$$\min_{\{U \subset V | s \in U, t \in \bar{U}\}} \sum_{e \in \langle U, \bar{U} \rangle} C(e),$$

where U is any subset of V that contains the source but not the sink, $\bar{U} = V - U$ is the set of nodes not in U, and $\langle U, \bar{U} \rangle$ denotes the set of edges that link a node in U to a node in $\bar{U}$. (The set of edges from any set U to $\bar{U}$ is referred to as a *cut* of the network since the removal of those edges separates U from the rest of the network.)

Claim 4 *The min-cut in a 1-commodity flow problem is an upper bound to the max-flow.*

Proof. The min-cut determines the minimum capacity needed to disconnect the source from the sink. The sum of the capacity of the edges in the cut can also be considered as the maximum flow that can pass through these edges. As these edges are a source sink cut, there cannot exist another path from the source to the sink that does not include these edges. Hence, it is an upper bound on the max-flow. **Q.E.D.**

Ford and Fulkerson The previous claim showed that the min-cut is the best max-flow value we could hope to achieve. Ford and Fulkerson proved that this value was always achievable in a single commodity flow problem.

Overview of the algorithm The algorithm works by initially setting the flow over every edge to 0, and then repeatedly finding an f-augmenting path. Using the f-augmenting path the flow can be increased by amount equal to the tolerance of the path. The algorithm terminates when there no longer exists any f-augmenting paths. We will now formalize these definitions and give a proof that the min-cut equals the max-flow in a single commodity flow problem.

Definition 5 Let f be a feasible flow over a network. An *f-augmenting path* is a source to sink path P, such that for each edge $e \in P$ the flow passing over the edge is less than the capacity of the edge.

Note: If an f-augmenting path does not exist in a network, then you have found a source sink cut. The reason is that there must exist a set of edges with no excess capacity, such that all paths from the source to the sink must include these edges. Otherwise the path that does not include these edges must be an f-augmenting path.

Definition 6 The *tolerance* of an f-augmenting path P is the minimum difference between the flow over any edge $e \in P$ and the capacity of e.

Claim 7 *The min-cut equals the max-flow in a single commodity flow problem*

Proof: Clearly the zero flow is a feasible flow in any network, given the definition of the capacity function. Let this flow be f. Find an f-augmenting path P over the network with flow f. This path allows us to increase f in amount equal to the tolerance of P. Continue to do this until no f-augmenting path can be found. We know this will occur after finitely many iterations. The reason for this is that there exists a finite number of edges, and after each iteration we max out at least one edge's capacity. Since there is no f-augmenting path, we must have found a source sink cut. The flow f must be equal to the capacity of this cut by construction. By Claim 4 the min-cut is an upper bound for the max-flow, and hence both must be optimal (f is the max-flow and the cut is the min-cut). **Q.E.D.**

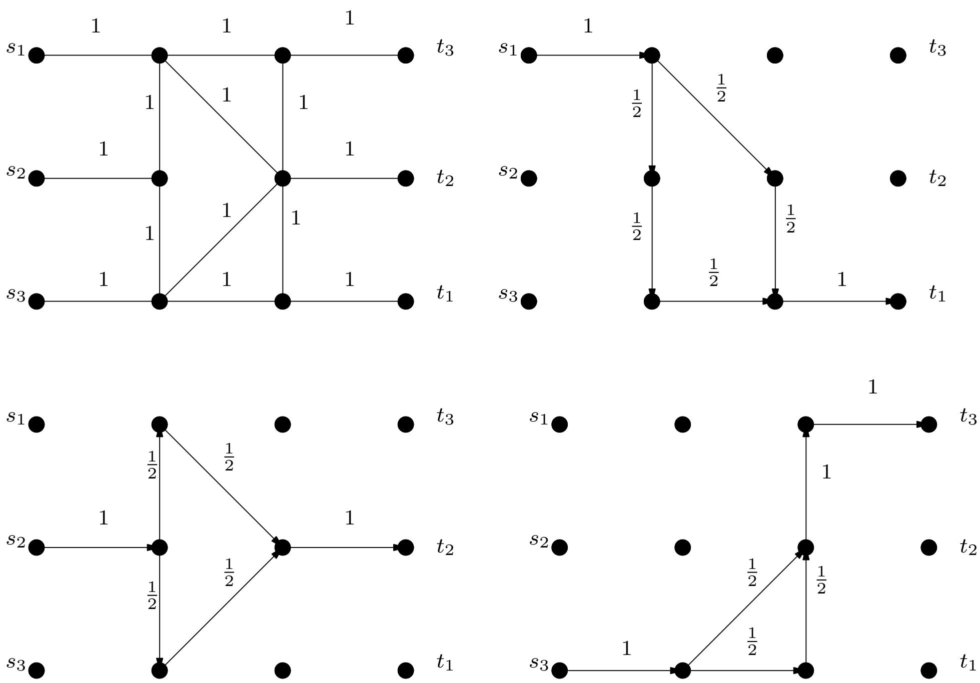

Figure 1: Solution to a 3-commodity flow problem.

2 Multicommodity flow

Definition 8 A *multicommodity flow problem* is a network that can be thought of as a 7-tuple (V, E, C, S, T, D, k), where

(1) V is a set of nodes

(2) E is a set of edges

(3) $C : E \to R_{\geq 0}$ is the capacity function for each edge

(4) S is the set of source nodes

(5) T is the set of sink nodes

(6) D is the set of demands

(7) k is the number of commodities.

Commodity i has source s_i, sink t_i, and demand D_i. The objective is to route D_i units for each commodity i from s_i to t_i, without violating any capacity constraints (see Figure 1.)

It may be impossible to simultaneously route D_i units for each commodity i, therefore, there are several definitions of max-flow for multicommodity flow problems. In the paper by Leighton and Rao they concentrate on a normalized version of max-flow. The justification for this choice is that the normalized version is representative of other definitions. This means that the techniques developed for this definition can be applied to alternative definitions as well.

Definition 9 The *normalized max-flow* in a multicommodity flow problem is the maximum value $f > 0$ such that fD_i units of commodity i can be routed simultaneously for each i, without violating any capacity constraints.

Definition 10 The *min-cut* in a multicommodity flow problem is the cut which minimizes the ratio of the capacity of the cut to the demand of the cut. More formally, the min-cut is

$$\varphi = \min_{U \subseteq V} \frac{C(U, \bar{U})}{D(U, \bar{U})},$$

where

$$C(U, \bar{U}) = \sum_{e \in \langle U, \bar{U} \rangle} C(e)$$

is the sum of capacities of the edges linking U to $\bar{U}$ and

$$D(U, \bar{U}) = \sum_{\{i | s_i \in U \wedge t_i \in \bar{U} \text{ or } t_i \in U \wedge s_i \in \bar{U}\}} D_i$$

is the sum of the demands whose source and sink are on opposite sides of the cut that separates U from $\bar{U}$. (Without loss of generality, we assume that the underlying graph is connected so that $C(U, \bar{U}) > 0$ for all U.)

It is important to note that this min-cut is a generalization of the min-cut in the single commodity flow setting. In addition, we can still establish the min-cut as an upper bound to the max-flow.

Claim 11 *The min-cut in a multicommodity flow problem is an upper bound to the max-flow.*

Proof: Let $\langle U, \bar{U} \rangle$ be our min-cut. Then let $i_1, i_2, ..., i_r$ be the commodities separated by our cut. Clearly all of the flow for these commodities must pass over our cut $\langle U, \bar{U} \rangle$ o.w. it would not be a cut. This implies

$$\begin{aligned} \sum_{j=1}^{r} f D_{i_j} &\leq C(U, \bar{U}), \\ \sum_{j=1}^{r} D_{i_j} &= D(U, \bar{U}), \\ f &\leq \frac{C(U, \bar{U})}{D(U, \bar{U})}. \end{aligned}$$

Q.E.D.

Before the paper by Leighton and Rao, the best known guarantee on the max-flow was a k-factor of the min-cut. Leighton and Rao make significant improvements on this bound. In doing this they are able to design the polynomial time algorithms mentioned in the introduction. First we will establish the k-factor. Then we will give the definition for the family of multicommodity flow problems for which Leighton and Rao establish their improved bound.

Claim 12 *For any multicommodity flow problem the max-flow is within a k-factor of the min-cut.*

Proof: Allow each commodity to have $1/k$ of the capacity for each edge in the min-cut. **Q.E.D.**

The work done by Leighton and Rao was established for uniform multicommodity flow problems. The reason for this choice is that these problems appear in many applications, such as NP-hard graph partitioning problems.

Definition 13 In a *uniform multicommodity flow problem* (UMFP) there is a commodity for every pair of vertices, and the demand for each commodity is the same. As all demands are equal, we will assume every demand is equal to one.

In a UMFP the demand across a cut $\langle U, \bar{U} \rangle$ is the product of the number of vertices in U, and $\bar{U}$.

$$D(U, \bar{U}) = |U||\bar{U}|$$
$$\varphi = \min_{U \subseteq V} \frac{C(U, \bar{U})}{|U||\bar{U}|}$$
$$\min_{U \subseteq V} \frac{|\langle U, \bar{U} \rangle|}{|U||\bar{U}|}.$$

Leighton and Rao show that for any multicommodity flow problem with uniform demands the max-flow is within an $O(\log n)$ factor of the min-cut. Furthermore, they show that this bound is tight by giving a set of graphs achieving this bound. This will be the topic of discussion for the next two sections.

3 A bad example

Since we have claimed that the $\log n$ bound is tight, let us start by giving the set of graphs that give this bound. Let G be a cubic graph where $|V| = n$ and the capacity of each edge is one. The important characteristic of this graph is that for any U subset of V we have $|\langle U, \bar{U} \rangle| \geq c \min\{|U|, |\bar{U}|\}$ for some constant c. It was established by Margulis that such graphs actually exist. By definition of G, we know that the min-cut of the corresponding UMFP is

$$\begin{aligned} \varphi &= \min_{U \subseteq V} \frac{|\langle U, \bar{U} \rangle|}{|U||\bar{U}|} \\ &\geq \min_{U \subseteq V} \frac{c}{\max\{|U|, |\bar{U}|\}} \\ &= \frac{c}{n-1}. \end{aligned}$$

Claim 14 *Max flow for UMFP is at most* $6/(n-1)(\log n - 2)$, *a* $\log n$ *factor smaller than the min cut.*

Proof: As G is 3-regular there is at most $n/2$ vertices within distance of $\log n - 3$ of any particular vertex v in V. The total number of commodities is $\binom{n}{2}$, and by above at least half of these have a shortest distance of $\log n - 2$ separating their source and sink. As we are in a UMFP we must have some flow value f for each commodity. Hence each commodity uses $f(\log n - 2)$ of the capacity available. With $\binom{n}{2}$ commodities, the total capacity of the graph must be at least $\frac{1}{2}\binom{n}{2}f(\log n - 2)$. As the graph is 3-regular the total capacity is $3n/2$. This implies

$$\begin{aligned} f &\leq \frac{3n}{\binom{n}{2}(\log n - 2)} \\ &= \frac{6}{(n-1)(\log n - 2)} \\ &\leq \frac{6\varphi}{c(\log n - 2)} \\ &= O\left(\frac{\varphi}{\log n}\right) \end{aligned}$$

Q.E.D.

Theorem 15 *For any n there is an n-vertex uniform multicommodity flow problem with max flow f and min-cut φ for which $f \leq O\left(\frac{\varphi}{\log n}\right)$.*

This theorem is proved by using the graphs of the previous proof. Basically take one of the graphs G, and then replace each edge in G with a path having some k vertices, so that after adding all these vertices to the edges in G we achieve the desired n. This leaves us with a UMFP that has a small min-cut. This min-cut must still have a $\log n$ factor greater than the max-flow. We will now establish that the $\log n$ factor is always achievable for UMFP.

4 Finding small cuts in UMFPs

We are finding a cut $\langle U, \bar{U} \rangle$ for any UMFP for which

$$\frac{C(U, \bar{U})}{|U||\bar{U}|} \leq O(f \log n),$$

where f is the max-flow of the UMFP. The quantity

$$\frac{C(U, \bar{U})}{|U||\bar{U}|}$$

is called the *ratio cost* of the cut $\langle U, \bar{U} \rangle$.

Theorem 16 *For any uniform multicommodity flow problem,*

$$\Omega\left(\frac{\varphi}{\log n}\right) \leq f \leq \varphi,$$

where φ is the min-cut of the UMFP.

The primal of a uniform multicommodity flow problem for a graph G is

$$\begin{aligned} \max \quad & f + \sum_{i \in \binom{V}{2}} \sum_{p \in \mathcal{P}_i} 0 \cdot f_p \\ \text{such that} \quad & \sum_{i=1}^{\binom{V}{2}} \sum_{p: e \in p \in \mathcal{P}_i} f_p \leq C(e) \text{ for every } e, \\ & f = \sum_{p \in \mathcal{P}_i} f_p \text{ for every commodity } i \in \tbinom{V}{2}, \\ & f_p \geq 0, \end{aligned}$$

where $\mathcal{P}_i$ is the set of paths connecting the commodity i.

The dual is

$$\begin{aligned} \min \quad & \sum_{e \in E} C(e) d(e) + \sum_{i \in \binom{V}{2}} 0 \cdot g_i, \qquad (1) \\ \text{such that} \quad & -g_1 - g_2 - \ldots - g_{\binom{V}{2}} = 1, \\ & \sum_{e \in p} d(e) \geq -g_i \text{ if } p \in \mathcal{P}_i, \\ & d(e) \geq 0 \text{ for every } e. \end{aligned}$$

An equivalent problem of the dual consists of the *distance constraint*,

$$\sum_{u,v \in V} d(u,v) \geq 1, \qquad (2)$$

where the sum is taken over all unordered pairs of nodes in G, and objective function,

$$W = \sum_{e \in E} C(e) d(e) = \vec{C} \cdot \vec{d},$$

which is called the *total weight* of the distance function d. We know $W = f$, however, non-zero distance edges do not give the min-cut in general.

Again, we are finding a cut $\langle U, \bar{U} \rangle$ for any UMFP for which

$$\frac{C(U, \bar{U})}{|U||\bar{U}|} \leq O(W \log n) = O(f \log n).$$

How can we bound the ratio cost

$$\frac{C(U, \bar{U})}{|U||\bar{U}|}?$$

We will partition our graph G so that the capacity of all the bridges between components is bounded by $O_\Delta(W \log n)$ which will bound $C(U, \bar{U})$, the numerator of the ratio cost (see Figure 4.)

Lemma 17 *For any graph G with arbitrary edge capacities, any $\Delta > 0$, and any distance function with total weight W it is possible to partition G into components with radius at most Δ so that the capacity of the edges connecting nodes in different components is at most*

$$\frac{4W \log n}{\Delta}.$$

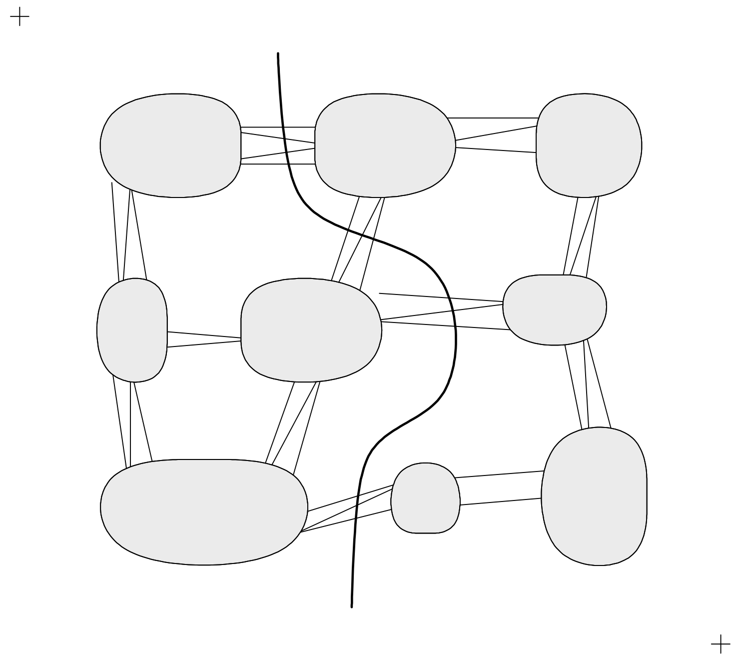

Figure 2: Lemma 17

We will put $\Delta = 1/2n^2$ and have

$$\frac{C(U,\bar{U})}{|U||\bar{U}|} \leq \frac{8n^2 W \log n}{|U||\bar{U}|}.$$

Note that a giant component could spoil the denominator of the ratio cost. (We will show the proof of this statement in a later section. We feel that accepting the truth of this statement will lead to an overall better understanding of the proofs that follow.)

Corollary 18 *For any graph G and any distance function with total weight W, we can either*

(1) find a component with radius $1/2n^2$ that contains at least $2/3$ of the nodes in G, or

(2) find a cut of G with ratio cost $O(W \log n)$.

Proof. Apply Lemma 17 with $\Delta = 1/2n^2$.If one of the components contains at least $2/3$ of $V(G)$, then we are done. We may assume that we can divide the components into two sets so that each set contains at least $n/3$. This cut has the capacity at most

$$\frac{4W \log n}{\Delta} = 8Wn^2 \log n.$$

Since both sides contain at leat $n/3$, the ratio cost of this cut is at most

$$\frac{8Wn^2 \log n}{(2n/3)(n/3)} = 36W \log n = O(W \log n).$$

Q.E.D.

In (2) of Corollary 18, everything's fine. But how can we resolve (1) with giant component? In fact, we are in better situation as follows.

Lemma 19 *For any graph G, if there is a distance function d with total weight W and a subset of nodes $T \subseteq V$ with $|T| \geq 2n/3$ and*

$$\sum_{u \in V-T} d(T,u) \geq \frac{1}{2n}, \tag{3}$$

then we can find a cut with ratio cost $O(W)$.

Proof. Coming soon... **Q.E.D.**

It turns out that the slight condition (3) in Lemma 19 is automatically satisfied (see the proof below.)

Lemma 20 *Given a graph G and a distance function d with total weight W that satisfies the distance constraint (2), we can find a cut with ratio cost $O(W \log n)$.*

Proof. Apply Lemma 17 with $\Delta = 1/2n^2$. By Corollary 18, we can either find a cut with ratio cost $O(W \log n)$ ((2) in the corollary) or we can find a component T with radius $1/2n^2$ that contains at least $2n/3$ nodes. We may assume that we are in the latter case. We want to apply Lemma 19 to find a cut with ratio cost $O(W)$. To do this we must show that the inequality (3),

$$\sum_{u \in V-T} d(T,u) \geq \frac{1}{2n}.$$

Since T has radius $1/2n^2$,

$$d(u,v) \leq d(T,u) + d(T,v) + \frac{1}{n^2}.$$

$$\begin{aligned}
\sum_{\{u,v\}} d(u,v) &= \frac{1}{2} \sum_{(u,v)} d(u,v) \\
&\leq \frac{1}{2} \sum_{(u,v)} \left(d(T,u) + d(T,v) + \frac{1}{n^2} \right) \\
&< n \sum_{u \in V-T} d(T,u) + \frac{1}{2}.
\end{aligned}$$

The distance constraint (2) implies

$$\sum_{u \in V-T} d(T,u) > \frac{1}{2n}.$$

Q.E.D.

This lemma implies Theorem 16, since $W = f$. Let's wind up this section with the involved partitioning algorithm to prove Lemma 17 and Lemma 19.

4.1 Proof of Lemma 17 and Lemma 19

Proof of Lemma 17: Let $C := C(E) := \sum_{e \in E} C(e)$ denote the *total capacity* on the edges of G. When $\Delta \leq \frac{4W \log n}{C(E)}$, the partition into singleton-components gives radius $0 \leq \Delta$ and the capacity of the edges running between different components $\leq C(E) \leq \frac{4W \log n}{\Delta}$.

If $\Delta > \frac{4W \log n}{C(E)}$, then we replace a second graph G^+ from G by replacing each edge e of G with a path of

$$\left\lceil \frac{C(E)d(e)}{W} \right\rceil = \left\lceil \frac{C(E)d(e)}{\vec{C} \cdot \vec{d}} \right\rceil \tag{4}$$

edges. Each edge along the path is assigned capacity $C(e)$ and distance 1. We will partition G by partitioning G^+. Note that in G^+ the distances are the (smallest) number of edges (not d).

Select $v \in V(G) \cap V(G^+)$. For each $i \geq 0$, define G_i^+ to be the induced subgraph of G^+ such that $V(G_i^+)$ is the set of vertices within distance i of v in G^+. Define

$$C(v) = C(G_0^+) := \frac{2C(E)}{n}. \tag{5}$$

Of course, $C(G_i^+) = C(E(G_i^+)) = \sum_{e^+ \in E(G_i^+)} C(e^+)$, $i > 0$. Let j denote the first $i \geq 0$ for which $(1+\epsilon)C(G_i^+) > C(G_{i+1}^+)$, i.e. the new edges added have small capacity $C(G_{i+1}^+) - C(G_i^+) < \epsilon C(G_i^+)$, where

$$\epsilon = \frac{W \log n}{\Delta C(E)} < \frac{1}{4}. \tag{6}$$

(There must be such a j since for large enough i, $G_{i+1}^+ = G_i^+ = G^+$.)

The induced subgraph G_j^+ form the first component of the partition. Repeat the same process for $G^+ - G_j^+$. The process is repeated until there are no longer any nodes $v \in G^+ \cap G$.

From the construction of G^+,

$$\begin{aligned} C(G^+) &= \sum_{e \in E} C(e) \left\lceil \frac{C(E)d(e)}{W} \right\rceil \leq \sum_{e \in E} C(e) \left(1 + \frac{C(E)d(e)}{W}\right) \\ &= \sum_{e \in E} C(e) + \frac{C(E)}{W} \sum_{e \in E} C(e)d(e) \\ &= 2C(E). \end{aligned} \tag{7}$$

The capacity of the edges leaving the component is at most $C(G_{j+1}^+ - G_j^+) = C(G_{j+1}^+) - C(G_j^+) < \epsilon C(G_j^+)$. (If $j = 0$, $C(G_1^+) < \epsilon C(G_0^+)$? They used this. I don't believe.) Since G_j^+ are disjoint, by (5) and (7), the total capacity on all edges leaving all components in G^+ is at most

$$\epsilon C(G^+) + \epsilon n C(G_0^+) \leq 2\epsilon C(E) + 2\epsilon C(E) = 4\epsilon C(E), \tag{8}$$

where the first term is for $j > 0$ and the second for $j = 0$.

The partition for G is derived from the components of G^+ in the natural way. The total capacity of the edges linking different components in G is at most $4\epsilon C(E) = \frac{4W \log n}{\Delta}$ (see (8)), as desired.

Let's show that each component has small radius. Consider a component with radius j in G^+. WMA $j > 0$. This component must have capacity

$$C(G_j^+) \geq (1+\epsilon)^j C(G_0^+) = (1+\epsilon)^j \frac{2C(E)}{n}$$

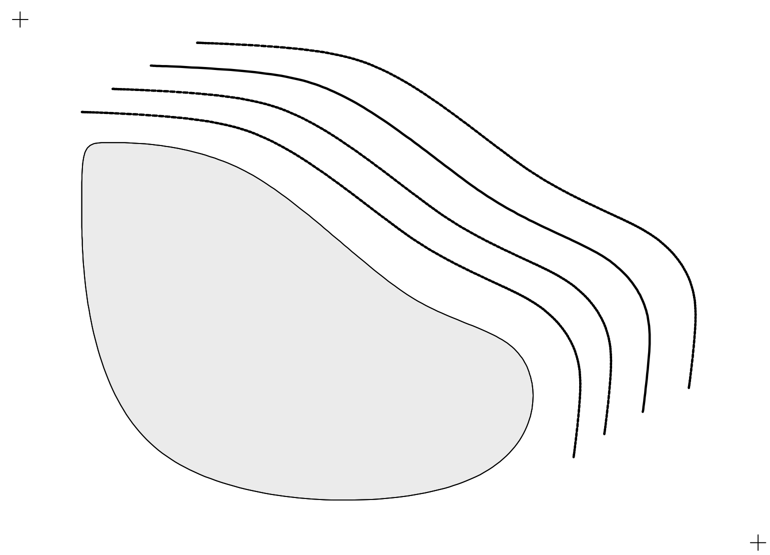

Figure 3: Dealing with giant component by equidistance curves.

by the definition. Of course, the capacity of the component is less than the total capacity of G^+:

$$(1+\epsilon)^j \frac{2C(E)}{n} \leq C(G_j^+) \leq C(G^+) \leq 2C(E)$$

and thus (since $\epsilon < 1/4$)

$$j \leq \frac{\log n}{\log(1+\epsilon)} \leq \frac{\log n}{\epsilon}.$$

Given a path of length l in G^+, the corresponding path in G has length at most $l\frac{W}{C}$ (see (4).) So the radius of each component in G is at most

$$\frac{W}{C(E)} \frac{\log n}{\epsilon} = \frac{W \log n}{C(E)\epsilon} = \Delta,$$

invoking the definition of ϵ (see (6).) **Q.E.D.**

Proof of Lemma 19: Let G_i^+ be the subgraph of G^+ within distance i of T. Define V_i to be the set of vertices of G corresponding to G_i^+, $n_i = |V - V_i|$, R_i to be the ratio cost of the cut $\langle V_i, V - V_i \rangle$ in G, and $R = \min R_i$ (see Figure 4.1.)

$$\sum_{u \in V-T} d_{G^+}(T, u) = \sum_{i \geq 0} n_i.$$

$$\sum_{i \geq 0} n_i \geq \frac{C}{W} \sum_{u \in V-T} d_G(T, u) \geq \frac{C}{2nW} \tag{9}$$

Since $|T| \geq 2n/3$, the capacity $C(V_i, V - V_i)$ is at least $R_i n_i (2n/3) \geq (2nRn_i/3)$.

$$\sum_{i \geq 0} \frac{2nRn_i}{3} \leq C(G^+) \leq 2C$$

$$R \leq \frac{3}{n \sum_{i \geq 0} n_i}$$

Invoking (9),

$$R \leq 6W = O(W).$$

Q.E.D.

Algorithm finding a small cut: Run the partitioning algorithm in Lemma 17 with $\Delta = 1/2n^2$. If there is no big component of at least 2/3, divide the components into two sets so that each set contains at least $n/3$ vertices. The cut gives $W \log n$ ratio cost (Corollary 18.) Otherwise, construct V_i from the big component T of at least 2/3 vertices and compute R_i's. Find a cut $\langle V_i, V - V_i \rangle$ giving $R = R_i$ which is less than W by Lemma 19.

5 Forwarding index problem

5.1 On general networks

Definition 21 A *uniform concurrent multicommodity flow* f on a graph G is a function $f : \mathcal{P} \rightarrow R_{\geq 0}$ such that, for any ordered pair of vertices (x, y) of G,

$$\sum_{p \in \mathcal{P}_{(x,y)}} f(p) = 1,$$

where $\mathcal{P}$ is the set of dipaths.

The edge congestion of an edge e of G is defined by

$$f(e) = \sum_{p \ni e} f(p),$$

and the *edge congestion* of the flow f is

$$\mu_f = \max_{e \in E} f(e)$$

Similarly, the vertex congestion of a vertex x is defined by

$$f(x) = \sum_{p \ni x} f(p),$$

where x is an internal vertex of path p, and the vertex congestion of the flow f is

$$m_f = \max_{x \in V} f(x).$$

Let's consider the integral multicommodity flow problem. That is, let f be an integral flow. Then the flow connecting a commodity is nothing but a path connecting them. A flow that minimizes μ_f (resp. m_f) is said to be *edge* (resp. *vertex*) *optimal*. In the optimal flow, μ_f (resp. m_f) is called the *edge* (resp. *vertex*) *forwarding index*. By applying the methods in [LR99], it is possible to bound the edge- and vertex-forwarding indices to within an $O(\log n)$-factor for every graph G.

5.2 Finding networks having the best possible forwarding indices

Let's consider the integral uniform concurrent multicommodity flow problem. ("If the best student got the average point, all the student got the same points")

Lemma 22 *Let G be a connected graph of order n. Then the vertex forwarding index of G is $\geq \sum_{(u,v)} d(u,v) - 1$ with equality holding if and only if there exists a flow of shortest paths for which the congestion of all vertices is the same.*

How can we construct a flow giving the same congestion everywhere if any? Consider the automorphism-vertex Latin square of $K_{3,3}$.

In general, such a Latin square guarantees the existence of the same congestion everywhere, that is, the best possible vertex forwarding index. ([HMS89], [HMOS94], and [SSZ02])

References

[HMS89] M.-C.Heydemann, J.-C.Meyer, D.Sotteau, On forwarding indices of networks, Discrete Appl. Math. 23 (1989) 103123.

[HMOS94] M.-C.Heydemann, J.-C.Meyer, J.Opatrný, D.Sotteau, Forwardingindices of consistent routings and their complexity, Networks 24 (1994) 7582.

[LR99] T.Leighton and S.Rao, Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms, Journal of the ACM 46 (1999) 787-832.

[M73] G.A.Margulis, Explicit constructions of concentrators, Prob. Inf. Trans. 9 (1973) 325- 332.

[SSZ02] S.Shim, J.Širáň, J.Žerovnik, Counterexamples to the uniform shortest path routing conjecture for vertex-transitive graphs, Discrete Appl. Math. 119 (2002) 281-286.

[W01] D.West, Introduction to graph theory, Upper Saddle River, NJ: Prentice Hall, 2001.