Multicommodity Flows

1 Introduction

In this lecture, we consider multicommodity flow problems and the relationships between maximum flows and minimum cuts in such networks. It is well-known that for a single commodity, the max-flow is equal to the min-cut; however, this does not generalize for multiple commodities. Here, we will discuss bounds developed by Leighton and Rao [LR1999] for a specific case which they call the uniform multicommodity flow problem. These bounds are quite useful in the sense that they have been used to design the first polynomial time approximation algorithms for more than two dozen NP-hard optimization problems.

The structure of the lecture is as follows. We begin by introducing notation and reviewing basic concepts of network flow. We then examine the known max-flow min-cut results for single commodity networks and provide an example to illustrate that the result does not hold in a multicommodity network. Next, we introduce the uniform multicommodity flow problem (UMFP) and state the only known (trivial) bound developed prior to [LR1999]. In the second section, we prove that in the UMFP, the max-flow is within a $\log n$ factor of the min-cut and show that this bound is tight. The proof is constructive in that it implies an efficient algorithm for finding such a cut. We conclude with a summary of the generalizations of the UMFP for which the $\log n$ factor has been shown to hold and a brief discussion of some hard problems for which the bounds can be employed to design efficient approximation algorithms.

2 Review of Network Flow and Known Bounds

We begin with notation and known results for both the single and multicommodity flow networks.

2.1 Single Commodity Networks

In the single commodity problem, we are given a graph $G = (V, E)$ with capacities C_e for each $e \in E$. (We assume for clarity of exposition that edges are undirected.) One node is designated as the source s and another as the sink t. We wish to route as much flow from s to t as possible without violating the capacity on any edge; this is known as the *max-flow*. A *min-cut* is the minimum capacity that must be removed to disconnect s from t.

Definition 1 *The min-cut in single commodity network $G = (V, E)$ is given by*

$$\min_{\{U \subset V | s \in U, t \in \bar{U}\}} \sum_{e \in \langle U, \bar{U} \rangle} C_e.$$

In this definition, U is any subset of V that contains the source but not the sink; $\bar{U} = V \backslash U$; and $\langle U, \bar{U} \rangle$ is the set of all edges linking a node in U with a node in $\bar{U}$. Clearly, the capacity of any cut is an upper bound on the max-flow since all flow from s to t must, by definition, cross the cut. The fundamental result of Ford and Fulkerson established that for a single commodity network, the max-flow is always equal to the min-cut.

Theorem 1 *In a single commodity network, the max-flow is equal to the min-cut. [due to Ford and Fulkerson, 1956]*

We refer the reader to [AMO1993] or to the original paper by Ford and Fulkerson [FF1956] for the proof. We will soon see that for the multicommodity case this result does not hold.

2.2 Multicommodity Networks

We now consider the multicommodity flow problem with $k \geq 1$ commodities. Multicommodity flow problems arise in a number of applications. For example, grains and other cargo items that are shipped by sea have unique origin-destination pairs, but these different commodities share common capacity on the ships. In addition, messages between different origins and destinations in a communications network also share capacity on the links. Each commodity i has demand D_i to be routed from source s_i to sink t_i. We wish to maximize the flow of the commodities such that the total flow of all commodities on any edge does not violate its capacity. However, since commodities are distinct and capacities are shared, we require a new definition of max-flow. Here we focus on a common definition known as *concurrent max-flow*, which can in some way be thought of as a fair max-flow. In the remainder of the discussion, we will use only the term *max-flow*.

Definition 2 *A concurrent max-flow in multicommodity network G is the maximum value of f such that fD_i units of commodity i can be simultaneously routed for each i without violating capacity constraints C_e on any edge.*

In multicommodity flow networks, the min-cut is defined as follows:

Definition 3 *The min-cut φ of a multicommodity flow network is given by*

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

where

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

is the sum of capacities of all the edges linking nodes in U with nodes in $\bar{U}$ 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 all demands with source and sink on opposite sides of the cut.

Again, the min-cut provides an upper bound for the max-flow.

Lemma 1 *In the multicommodity flow problem, $f \leq \varphi$, where f is the max-flow and φ is the min-cut.*

Proof. Let $i_1, i_2, \ldots, i_r$ be the commodities with source and sink on opposite sides of some cut $\langle U, \bar{U} \rangle$. All flow for these commodities must cross the cut. Thus,

$$\sum_{j=1}^{r} f D_{i_j} \leq C(U, \bar{U}).$$

We know that

$$\sum_{j=1}^{r} D_{i_j} = D(U, \bar{U}),$$

which implies

$$f \leq \frac{C(U, \bar{U})}{D(U, \bar{U})} = \varphi.$$

□

However, the network in Figure 1 illustrates that this bound cannot be satisfied at equality for all multicommodity flow networks. Indeed, for general networks, only a trivial relationship is known between the max-flow and min-cut.

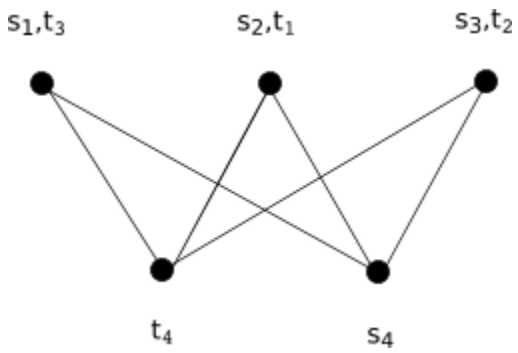

Figure 1: All demands and capacities are 1. The max-flow is 3/4 and the min-cut is 1.[OS1981]

Lemma 2 *In general multicommodity flow networks with k commodities, the max-flow is within a factor of k of the min-cut.*

Proof. Solve the max-flow problem separately for each commodity, using $\frac{1}{k}C(e)$ as the capacity of each edge. □

In the remainder of this lecture, we will focus on the results of Leighton and Rao for a particular multicommodity flow problem and find that for these networks, a much tighter bound is possible.

3 Max-Flow Min-Cut Theorems for Multicommodity Flow

We proceed by defining the uniform multicommodity flow problem (UMFP) that is the subject of Leighton and Rao's work and then present their main results.

3.1 Uniform Multicommodity Flow Problem

The uniform multicommodity flow problem has a commodity for every pair of nodes in $G = (V, E)$, with equal demand for every commodity (without loss of generality, the demand may be assumed to be one). The capacities and network structure are arbitrary, but edges are assumed undirected. This is the primary setting in which Leighton and Rao have established their results and the one on which we focus here. The results are extended to multicommodity flows on directed graphs, flow problems in which paths are required to be short, and problems in which the uniform demand assumption is relaxed; the reader is referred to the original paper for a treatment of these topics.

In the UMFP, we are able to simplify some of the previous definitions.

Definition 4 *The demand across any cut $\langle U, \bar{U}\rangle$ in the UMFP is given by*

$$D(U, \bar{U}) = |U||\bar{U}|,$$

that is, the product of the number of nodes in U and that in $\bar{U}$.

Definition 5 *The min-cut in the UMFP is given by*

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

where

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

If all capacities are one, this reduces to

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

The result that we are going to prove is that for UMFP $\varphi \geq f = \Omega(\frac{\varphi}{\log n})$, and the $\log n$ bound cannot be improved in general. For this result we must show a family of graphs such that $f = O(\frac{\varphi}{\log n})$, the upper bound, and then show the general lower bound that $f = \Omega(\frac{\varphi}{\log n})$. Note that this result can be extended to a polynomial time algorithm for finding such a cut.

3.2 Upper Bound

Theorem 2 *There exists a family of graphs for arbitrarily large n such that $f = O(\frac{\varphi}{\log n})$.*

Proof. It is known that for small $\epsilon > 0$ there exist families of 3-regular graphs with unit capacities such that on n nodes they satisfy $\left|\langle U, \bar{U}\rangle\right| \geq \epsilon \min\{|U|, |\bar{U}|\}$ for all $U \subset V$. Then for any such graph we have that

$$\varphi = \min_{U \subset V} \frac{\left|\langle U, \bar{U}\rangle\right|}{|U|\,|\bar{U}|} \geq \min_{U \subset V} \frac{\epsilon \min\{|U|, |\bar{U}|\}}{|U|\,|\bar{U}|} = \min_{U \subset V} \frac{\epsilon}{\max\{|U|, |\bar{U}|\}} = \frac{\epsilon}{n-1}$$

Consider any node $v \in V$, since G is 3-regular we know that there are at most $1 + 3 \cdot (2^k - 1)$ nodes within distance k of v. For $k = \log n - 3$ we have that there are at most

$$1 + 3 \cdot (2^{\log n - 3} - 1) = 3 \cdot (n 2^{-3}) - 2 \leq \frac{3n}{8} \leq \frac{n}{2}$$

nodes of distance $\log n - 3$ from v. Since v has a commodity demand with every other node we have that half of the commodities paired with v must use capacity on at least $\log n - 2$ edges along any route of the flow. As this is true for any node $v \in V$ and f is the flow of each commodity we know that the total amount of capacity consumed by a flow f must be at least

$$\frac{1}{2}\binom{n}{2} f(\log n - 2) \leq \frac{3n}{2},$$

where $\frac{3n}{2}$ is clearly the capacity for our graph.

Therefore we get that

$$f \leq \frac{3n}{\binom{n}{2}(\log n - 2)} \leq \frac{3n(n-1)\varphi}{\epsilon\binom{n}{2}(\log n - 2)} = O\left(\frac{\varphi n^2}{n^2 \log n}\right) = O\left(\frac{\varphi}{\log n}\right).$$

□

3.3 Lower Bound

For the lower bound we find a cut U which satisfies

$$f = \Omega\left(\frac{C(U, \bar{U})}{|U||\bar{U}| \log n}\right).$$

To obtain this cut we will utilize the dual of (UMFP), in which we can think of having a distance function on the edges, $d : E \to \mathbf{R}^+$ where we want to minimize $(C^T d)$. This distance function will have to satisfy the constraint

$$\sum_{v_i, v_j \in V : i < j} d_{v_i, v_j} \geq 1$$

where we use shorthand for standard shortest distance between two nodes using our distance function. Linear programming strong duality will give us such a distance function with $f = (C^T d)$.

Once we have the distance function d we will partition the graph into regions of small radius with certain properties which allow us to use a Ramsey type argument, we always find at least one of two properties, both of which ensure we find a cut of desired size.

Lemma 3 *We can partition the graph into components of radius at most $\frac{1}{2n^2}$ such that the total sum of capacities on all edges going between different components is at most $6(C^T d)n^2 \log n$.*

Proof. If $\frac{1}{2n^2} \leq \frac{6(C^T d)n^2 \log n}{(C^T \mathbf{1})}$ then using the trivial n partion clearly satisfies our conditions. So we may assume $\frac{1}{2n^2} > \frac{6(C^T d)n^2 \log n}{(C^T \mathbf{1})}$.

First we will modify G by taking each edge e and subdividing it $\left\lceil \frac{(C^T \mathbf{1})d_e}{(C^T d)} \right\rceil - 1$ times, and each new edge will have the same original capacity.

Now we will choose our partitions, starting by choosing an arbitrary node $v \in V$.

Let $P_0 = \frac{(C^T \mathbf{1})}{n}$, and for all $i > 0$ define $P_i = \sum_{e_i \in E(G_i)} C_{e_i}$ where G_i is the graph induced from G by taking all nodes within cardinal distance i of v. Let j be the minimum subscript such that

$$P_{j+1} < \left(1 + \frac{2(C^T d)n^2 \log n}{(C^T \mathbf{1})}\right) P_j$$

which clearly exists since the graph is finite.

The partition which includes v will be all of the original nodes which are in G_j. Delete all of these nodes from the original graph and repeat starting from subdividing the edges.

Note that $P_{j+1} - P_j$ is the sum of the capacities leaving G_j. This implies that the capacity of edges leaving a partition is at most

$$\frac{2(C^T d)n^2 \log n}{(C^T \mathbf{1})} P_j.$$

Since we subdivided the original edges each of the original capacities could have been counted at most twice in any P_i and never in more than one P_i. Therefore we get a weak upper bound on the total sum of edges leaving partitions as

$$\frac{2(C^T d)n^2 \log n}{(C^T \mathbf{1})} \left(2(C^T \mathbf{1}) + nP_0\right) = \frac{2(C^T d)n^2 \log n}{(C^T \mathbf{1})} \left(2(C^T \mathbf{1}) + (C^T \mathbf{1})\right) = 6(C^T d)n^2 \log n$$

as desired.

Thus we need only show the partitions have small enough radius. Note we may assume $j > 0$ since partitions for $j = 0$ are trivial.

We wish to have an upper bound on how often we possibly count each edge capacity, so we see that we count for capacity at most

$$\sum_{e \in E} C_e \frac{(C^T 1)d_e}{(C^T d)} \leq \sum_{e \in E} C_e + \frac{(C^T \mathbf{1})}{(C^T d)} \sum_{e \in E} C_e d_e = 2(C^T \mathbf{1}).$$

Thus for any partition we have that

$$P_j \leq \left(1 + \frac{2(C^T d)n^2 \log n}{(C^T \mathbf{1})}\right)^j \frac{2(C^T \mathbf{1})}{n} \leq 2(C^T \mathbf{1}).$$

Note that $\frac{2(C^T d)n^2 \log n}{(C^T \mathbf{1})} < \frac{1}{3}$ and take a logarithm to get that

$$j \leq \frac{\log n}{\log\left(1 + \frac{2(C^T d)n^2 \log n}{(C^T \mathbf{1})}\right)} \leq \frac{\log n}{\frac{2(C^T d)n^2 \log n}{(C^T \mathbf{1})}}.$$

However, j is the radius in the subdivision, so the radius of the component is at most

$$j \frac{(C^T d)}{(C^T \mathbf{1})} \leq \frac{(C^T d) \log n}{(C^T \mathbf{1}) \frac{2(C^T d)n^2 \log n}{(C^T \mathbf{1})}} \leq \frac{1}{2n^2}.$$

□

We continue with this partition to immediately obtain a Ramsey type result where one case of two will satisfy our lower bound.

Corollary 1 *Either some component contains $\frac{2}{3}n$ nodes or we can find a desired cut of ratio $O((C^T d)\log n)$.*

Proof. Assume we do not find a component containing $\frac{2}{3}n$, then clearly we can group the components into a partition of precisely two sets of size at least $\frac{n}{3}$.

Since the cost of the capacities leaving one of the partitions is clearly less than the cost from the original partition we get that the cut defined by this partition is at most $6(C^T d)n^2 \log n$. The ratio is thus at most

$$\frac{6(C^T d)n^2 \log n}{\frac{n}{3}\frac{n}{3}} = O\left((C^T d)\log n\right)$$

□

Thus we only need to handle the case when our partition gave us some component of size at least $\frac{2}{3}n$.

Lemma 4 *If some component T contains $\frac{2}{3}n$ nodes, then we can find a cut of size $O((C^T d))$.*

Proof. Let n_i be the number of nodes from the subdivided graph not within distance i of T, where we denote the cardinal distance from a node v to T by $h_{v,T}$.

We easily get that

$$\sum_{i=0}^{\infty} n_i = \sum_{v \in V-T} h_{v,T}.$$

To get a bound on $\sum_{v\in V-T} d_{v,T}$ we use the diameter of T and the constraint from the dual to observe that

$$1 \le \sum_{u,v\in V} d_{u,v} \le \sum_{u,v\in V: u\neq v} \left(d_{u,T} + d_{v,T} + 2\frac{1}{2n^2}\right) \le (n-1)\left(\sum_{v\in V-T} d_{v,T}\right) + \frac{\binom{n}{2}}{n^2} < n\left(\sum_{v\in V-T} d_{v,T}\right) + \frac{1}{2}$$

and get the bound

$$\sum_{v\in V-T} d_{v,T} > \frac{1}{2n}.$$

This gives us that

$$\sum_{i=0}^{\infty} n_i = \sum_{v\in V-T} h_{v,T} \ge \frac{(C^T\mathbf{1})}{(C^T d)} \sum_{v\in V-T} d_{v,T} \ge \frac{(C^T\mathbf{1})}{2n(C^T d)}$$

Let U_i be the cut determined by all nodes within distance i of T. Let

$$R = \min_i \frac{C(U_i, \bar{U}_i)}{|U_i| n_i},$$

then we have that the capacity of any cut is at least $\sum_{i=0}^{\infty} R_i n_i |U_i| \ge \sum_{i=0}^{\infty} \frac{2nRn_i}{3}$. This gives us, with use of the earlier bound on the capacities, that

$$\frac{R2n}{3}\sum_{i=0}^{\infty} n_i \le 2(C^T\mathbf{1}).$$

Thus we have

$$R \le \frac{3(C^T\mathbf{1})}{n\sum_{i=0}^{\infty} n_i} \le \frac{3(C^T\mathbf{1})}{n\frac{(C^T\mathbf{1})}{2n(C^T d)}} = 6(C^T d) = O((C^T d)).$$

□

Corollary 2 *Any instance of UMFP satisfies $\Omega\left(\frac{\varphi}{\log n}\right) = f \le \varphi$.*

Proof. Follows directly from the preceding results of this section. □

The algorithm follows closely to the proof. First solve the dual to find the distance function and the optimal value, then partition the graph as described. If no component has $\frac{2n}{3}$ nodes finding the cut is easy, otherwise add the closest node to the component T one at a time and check each of the n_0 cuts and take the best ratio.

4 Applications to Approximation Algorithms

Leighton and Rao [LR1999] go on to show that their results still hold for certain generalizations of the UMFP and that they are useful for designing efficient approximation algorithms for many NP-hard optimization problems.

Relaxing the restriction that graphs be undirected, the authors show that the max-flow is within a log n factor of the min-cut for directed UMFP's with n nodes. With minor modifications to the proofs, the uniformity of demands between all pairs of nodes can also be relaxed. In a *Product Multicommodity Flow Problem (PMFP)*, a weight is associated with each node and the demand for the commodity between each pair of nodes i and j is the product of their weights. For an undirected PMFP with k commodities, the max-flow is always within a log k factor of the min-cut. This result extends to directed PMFP's.

The *log n* result enabled these authors and many others to develop efficient approximation algorithms for more than two dozen NP-hard optimization problems. Among these are a number of graph partitioning problems. Here, we mention only a few.

The *sparsest cut* φ of $G = (V, E)$ is a partition of the vertices that satisfies

$$\varphi = \min_{\{U \subset V\}} \frac{\langle U, \bar{U} \rangle}{|U||\bar{U}|},$$

where $\langle U, \bar{U} \rangle$ is the set of edges connecting vertices on one side of the partition with vertices on the other. Finding the sparsest cut is NP-hard, but it can clearly be approximated to within an $O(\log n)$ factor by casting the problem as a UMFP (all capacities equal to 1 and demands of 1 between all pairs of nodes) and finding a cut with ratio cost $O(f \log n)$.

Other problems for which the result in [LR1999] led to an efficient approximation algorithm include finding balanced cuts and directed cuts, minimum cut linear arrangement, minimum feedback arc set, crossing number, and minimum VLSI circuit layout, among others. Therefore, although the classes of multicommodity flow problems for which the logarithmic bound holds seem restrictive at first glance, the result is actually quite useful for many other applications.

5 Related Work and Conclusions

Although the paper by Leighton and Rao from which these results are taken appeared in 1999, the $\log n$ bound was shown much earlier and appeared in various conference proceedings. It is a seminal result in the study of multicommodity flow problems. Since its publication, other advances have been made. We briefly mention a few.

Garg and Könemann [GK1998] find a polynomial time algorithm to obtain good approximations for the maximum concurrent multicommodity flow. The running time for their approach depends on the time to compute a single commodity minimum cost flow in a graph with non-negative edge weights. Fleischer [F1999] improves the running time of this algorithm for sparse graphs or instances with $k > m/n$, where k is the number of commodities, m is the number of edges, and n is the number of vertices. Finally, Batra, et al [BGG2005] study the maximum multicommodity flow problem, in which the objective is to maximize the total flow (as opposed to the concurrent flow that we have previously considered). They improve upon the running times of approaches presented in [GK1998, F1999] by reducing the total number of shortest path computations needed, and they find a $1 + \epsilon$ approximation to the maximum multicommodity flow.

In summary, finding maximum multicommodity flows and concurrent flows is much more challenging than solving single commodity problems. The well-known max-flow min-cut result for the single commodity case does not generalize. Leighton and Rao's work proved that for a number of classes of the problem, the max-flow is within a $\log n$ factor of the min cut. This led to approximation algorithms for numerous hard and seemingly unrelated problems, and it contributed to future advances in approximation algorithms for the maximum multicommodity flow problem itself.

6 References

[AMO1993] Ahuja, R.K., T.L. Magnanti, and J.B. Orlin (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, NJ.

[BGG2005] Batra, Garima, Naveen Garg, and Garima Gupta(2005). "Heuristic improvements for computing maximum multicommodity flow and minimum multicut." *Lecture Notes in Computer Science*, **3669**.

[FF1956] Ford, L.R. and D.R. Fulkerson (1956). "Sur le probléme des courbes gauches en topologie." *Canadian Journal of Mathematics*, **8**, 399-404.

[GK1998] Garg, Naveen, and Jochen Könemann (1998). "Faster and simpler algorithms for multicommodity flow and other fractional packing problems." *IEEE Symposium on Foundations of Computers Science*, 300-309.

[LR1999] Leighton, T. and S. Rao (1999). "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms." *Journal of the ACM*, **46(6)**, 787-832.

[OS1981] Okamura H. and Seymour, P.D. (1981). "Multicommodity flows in planar graphs." *J. Combin. Theory, Ser. B.*, **31**, 75-81.