Multi-Commodity Max-Flow Min-Cut Theorems and Applications

David Wajc
Last Updated: May 3, 2019

Abstract

In these notes we present an overview of some classic (approximate) multi-commodity max-flow min-cut theorems and some of their applications to approximation algorithms.

1 Introduction

Consider a single-commodity network flow instance, $N = (G, s, t, c)$. The maximum amount of flow which can be sent from the source s to the sink t is clearly upper bounded by the capacity of the minimum cut separating s and t. The celebrated Maximum-Flow Min-Cut Theorem of Ford and Fulkerson [4] (also proven in the same year by Dantzig and Fulkerson [2] and Elias, Feinstein and Shannon [3]) asserts that these quantities coincide. This fundamental theorem has found myriad applications over the years, including simple proofs of classic combinatorial theorems such as Menger's, Hall's, König's and Dilworth's theorems, as well as countless algorithmic applications. One might therefore wonder which generalizations of maximum flow admit similar max-flow min-cut theorems and what applications they may yield.

A natural generalization of the maximum flow problem to consider is *multi-commodity* flow problems. One such generalization is the *concurrent* multi-commodity flow, where we have source-sink pairs $\{(s_i, t_i)\}_{i=1}^k$ and *demand* D_i for each pair (s_i, t_i). This problems asks to send the maximum common fraction f of each demand D_i along the set of paths $P_i = \{p : s_i \rightsquigarrow t_i\}$, while respecting the edge capacities, c_e.[1] (Single-commodity flow is the special case of a single source-sink pair with demand one.) Written as an LP, the concurrent multi-commodity flow problem and its dual are the following.[2]

Primal			Dual		
maximize	f		minimize	$\sum_e l_e \cdot c_e$	
subject to:	$\sum_{p \in P_i} x_p \geq f \cdot D_i$	$\forall i$	subject to:	$\sum_i D_i \cdot d(s_i, t_i) \geq 1$	
	$\sum_{p \ni e} x_p \leq c_e$	$\forall e \in E$		$\sum_{e \in p} l_e \geq d(s_i, t_i)$	$\forall p \in P_i$
	$x_p \geq 0$	$\forall i, \forall p \in P_i$		$d(s_i, t_i), l_e \geq 0$	$\forall i, e$

Figure 1: The Concurrent Multi-commodity Flow LP and its dual

The dual variable names are somewhat suggestive. We think of the l_e as edge lengths, and the $d(s_i, t_i)$ variables as the length of the shortest path (according to these lengths) between s_i and t_i.

As in the single-commodity flow problem, the max flow is bounded by the (appropriate generalization of) min cut. Here we define the *min-cut* as follows. For a cut $(U, \bar{U})$, we denote the capacity of edges crossing this cut by $C(U, \bar{U}) := \sum_{e \in (U, \bar{U})} c_e$ and the demand separated by it by $D(U, \bar{U}) := \sum_{i : |\{s_i, t_i\} \cap U| = 1} D_i$. The *sparsity* of a cut $(U, \bar{U})$ is $R(U, \bar{U}) := \frac{C(U, \bar{U})}{D(U, \bar{U})}$. The *min-cut* (sparsest cut) is simply the minimum such ratio over all cuts separating source-sink pairs,

$$R^* := \min_{U \subseteq V, D(U, \bar{U}) \neq 0} R(U, \bar{U}),$$

[1] Another generalization is the *sum* multi-commodity flow problem, where we have source-sink pairs $\{(s_i, t_i)\}_{i=1}^k$ and we wish to maximize the amount of flow sent between these source-sink pairs along the set of paths $P_i = \{p : s_i \rightsquigarrow t_i\}$ connecting the pairs s_i and t_i, again while respecting the edge capacities, c_e. We will not discuss this problem in great depth here, though.

[2] Note that this LP generalizes the path-form LP for max single-commodity flow we saw in class before.

We first verify that the min-cut is indeed an upper bound on the max flow. Let $i_1, i_2, \ldots, i_r$ denote the set of source-sink pairs separated by some cut $(U, \bar{U})$. Since all flow between these source-sink pairs must cross this cut and the flow must respect the capacities across the cut, we have for a flow of value f that $\sum_{j=1}^{r} f \cdot D_{i_j} \leq C(U, \bar{U})$. But as $D(U, \bar{U}) = \sum_{j=1}^{r} D_{i_j}$, this implies that any flow has value at most $f \leq C(U, \bar{U})/D(U, \bar{U}) = R(U, \bar{U})$. This bound holds for the max flow, f^*, and $(U, \bar{U})$ the min cut, and so the max multicommodity flow is upper bounded by the min-cut, as for the single-commodity case. However, unlike for single-commodity flow, these values need not coincide. In fact, these values can be as far as an $\Omega(\log n)$ multiplicative factor apart. In this lesson we will show that that is the most these terms can differ by.

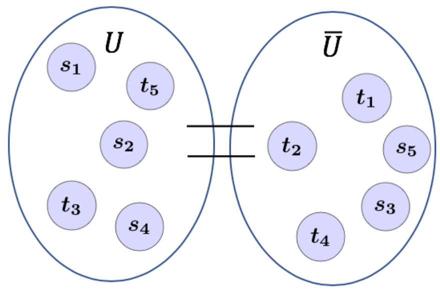

Figure 2: Illustration of cut $(U, \bar{U})$ proving that $f \leq C(U, \bar{U})/D(U, \bar{U}) = 2/5$.

2 Undirected Concurrent Flow with Uniform Demands

For simplicity, we will focus on undirected concurrent multi-commodity flow with *uniform demands*, first studied in the seminal work of Leighton and Rao [5]. Here *all* node pairs (u, v) are source-sink pairs, and they all have the same demand. By normalizing appropriately, we may assume this common demand is one. As Leighton and Rao showed, even for this simple case of multi-commodity flow there exist instances with a logarithmic gap between max flow and the min cut.

Theorem 2.1 ([5])**.** *There exist n-node graphs for which the uniform concurrent multi-commodity maximum flow and min cut values differ by a $\Omega(\log n)$ factor.*

As we shall see, $\Theta(\log n)$ is the correct answer: for every network the max flow is always at least $\Omega(1/\log n)$ times the min-cut. We will prove this theorem in the remainder of this section. In Section 3, we shall see applications of this result.

One component we will rely on in our proof of our approximate min-cut max-flow theorem are low-diameter decompositions as we saw in class before, generalized to *weighted graphs*. A low-diameter decomposition of a graph G with capacities c_e and edge lengths l_e (think of a dual solution to the LP) is a partition of the graph G into parts of low diameter (according to these l_e) which "cuts" few edges. The properties we need of low-diameter decompositions are captured by the following lemma. (These properties are obtained by the low-diameter decomposition using exponential delay we saw in class before, by generalizing to weighted graphs appropriately.)

Lemma 2.2 (Low-Diameter Decompositions)**.** *Given parameter β and undirected graph $G = (V, E)$ with edge lengths $\{l_e\}_e$ and capacities $\{c_e\}_e$, one can compute in polytime a partition of G into connected components such that with probability at least $1/4$,*

1. *Each part has l-diameter at most $\log n/\beta$.*

2. *The capacity of the edges $E' \subseteq E$ separated is at most $\sum_{e \in E'} c_e \leq 2\sum_e c_e \cdot l_e \cdot \beta$.*

We now return to the concurrent multi-commodity flow LP and its dual. Let f^* be the max flow value. Consider some optimal dual solution with variables l_e and $d(s_i, t_i)$. As observed before, we can set the $d(s_i, t_i)$ to be as high as the minimum l-length path from s_i to t_i. As we consider the uniform multi-commodity problem where all node pairs have demand of one, the LP constraints imply that

$$\sum_{u,v} d(u,v) \geq 1. \tag{1}$$

Let $f^* = \sum_e l_e \cdot c_e$ be the dual solution's value (and the max flow value). We will show a cut with sparsity at most $O(f^* \log n)$, implying our claimed bound on the flow-cut gap. We start with a simple application of Lemma 2.2 to potentially finding a cut of sparsity at most $O(f^* \log n)$.

Lemma 2.3. *For any graph G with lengths $\{l_e\}$, capacities $\{c_e\}$ and with $f^* = \sum_e c_e \cdot l_e$, we can either*

1. *find a connected component with l-diameter $1/2n^2$ that contains at least $n/3$ nodes of G, or*
2. *find a cut in G with sparsity $O(f^* \cdot \log n)$.*

Proof. We run the algorithm of Lemma 2.2 with parameter $\beta = 2n^2 \cdot \log n$. If one of the components contains at least $n/3$ of the nodes of G, we are done (as the parts have l-diameter at most $\log n/\beta = 1/2n^2$). Otherwise, we group the parts into two sides of a cut, both of size at least $n/3$.[3] By Lemma 2.2, we know that the capacity of edges cut by the partition is at most $2\sum_e c_e l_e \cdot \beta = 4f^* n^2 \log n$. But as the two sides of the partition have at least $n/3$ nodes, the sparsity of this cut is at most

$$\frac{4f^* n^2 \log n}{(n/3)(n/3)} = 36 f^* \log n = O(f^* \log n). \qquad \square$$

Now, the second condition of Lemma 2.3 is exactly what we want, while the first condition seems a little cryptic. We will show that it will also allow us to find a cut with sparsity $O(f^*) \leq O(f^* \log n)$.

Lemma 2.4. *Let $T \subseteq V$ be a set of $|T| \geq n/3$ vertices with l-diameter at most $1/2n^2$. Then we can find a cut with sparsity at most $O(f^*)$.*

To prove the above lemma, we will need the following lemma, which asserts that a ball of small diameter must be relatively far away from nodes outside of this ball.

Lemma 2.5. *Let $T \subseteq V$ be a set of vertices with l-diameter at most $1/2n^2$. Then $\sum_{u \in V \setminus T} d(T, u) \geq \frac{1}{2n}$.*

Proof. By triangle inequality and the diameter bound we have that for each pair of vertices u, v we have $d(u,v) \leq d(T,u) + 1/2n^2 + d(T,v)$. By Equation (1) and the above we find that

$$\begin{aligned} 1 &\leq \sum_{u,v} d(u,v) \\ &\leq \sum_{u,v} \big(d(T,u) + 1/2n^2 + d(T,v)\big) \\ &< n \cdot \left(\sum_{u \in V \setminus T} d(T,u) \right) + \frac{1}{2}. \end{aligned} \qquad \square$$

[3] We can do this by starting with an empty group U and adding parts to U until its size first exceeds $n/3$ due to addition of some part P. This means the group previously had size less than $n/3$ and together with P (which has size at most $n/3$). So, we find that this group has size at most $|U \setminus P| + |P| \leq 2n/3$, and so its complement also has size $|\bar{U}| = n - |U| \geq n - 2n/3 = n/3$.

The above distance constraint for vertices outside of T will allow us to show the complementary result to Lemma 2.3 — a procedure to compute a sparse cut in the case that one component with many nodes has low l-diameter.

Proof of Lemma 2.4. We expand every edge e of G into a path of length $\lceil \frac{C \cdot l_e}{f^*} \rceil$, where $C = \sum_e c_e$ is the total graph capacity and $f^* = \sum_e c_e \cdot l_e$ is the dual solution's value. We assign each edge in the path replacing edge e a capacity of c_e. We refer to this expanded graph as G^+. (Expanding edges this way is mainly done to simplify discussion and allow us to think about distances as integer values.) By its definition we find that the capacity of G^+ is not much larger than that of G. Specifically, it is at most

$$C(G^+) = \sum_e c_e \cdot \left\lceil \frac{C \cdot l_e}{f^*} \right\rceil \leq \sum_e c_e \cdot \left(1 + \frac{C \cdot l_e}{f^*}\right) = 2C, \tag{2}$$

where we rely on $\sum_e c_e \cdot l_e = f^*$ in the last step.

Moreover, as every path p in G is replaced by a path in G^+ of length at least C/f^* times the l-length of p. Consequently, distances in G^+ are at least C/f^* times larger than their counterparts in G. In particular, we have that

$$d_{G^+}(T, u) \geq \frac{C}{f^*} \cdot d_G(T, u), \quad \forall u \in V \setminus T. \tag{3}$$

Now, let $V_i \supseteq T$ be the set of nodes in G at distance at most i from T in G^+, and let $n_i := |V - V_i|$ be the number of nodes at distance greater than i from T in G^+. By definition of n_i and integrality of the distances in G^+, we have that

$$\sum_{e \in V \setminus T} d_{G^+}(T, u) = \sum_{i \geq 0} n_i. \tag{4}$$

(To see this, note that a node at distance d from T is counted in $n_0, n_1, n_2, \ldots, n_{d-1}$, or d of the summands of the right hand side.)

Combining inequalities 3 and 4 together with the lower bound of Lemma 2.5, we obtain

$$\sum_{i \geq 0} n_i \overset{4}{=} \sum_{e \in V \setminus T} d_{G^+}(T, u) \overset{3}{\geq} \frac{C}{f^*} \cdot \sum_{e \in V \setminus T} d_G(T, u) \overset{2.5}{\geq} \frac{C}{2nf^*}. \tag{5}$$

As every cut $(V_i, V \setminus V_i)$ has at least $|V_i| \geq |T| \geq n/3$ nodes on one side and n_i on the other, and as every edge of G^+ is counted in exactly one of these cuts, we find that the ratio of capacities to demands of these cuts is at most

$$\frac{\sum_i C(V_i, V \setminus V_i)}{\sum_i |V_i||V \setminus V_i|} \overset{2,\,|V_i| \geq n/3}{\leq} \frac{2C}{(n/3) \sum_i n_i} \overset{5}{\leq} 3f^*. \tag{6}$$

But as at least one of the cuts $(V_i, V \setminus V_i)$ must have sparsity (or ratio of capacity to demands separated) at most this average, we find that one of these cuts has sparsity at most $3f^* = O(f^*)$. As there are only polynomially-many distinct cuts $(V_i, V \setminus V_i)$ as above, this existence claim can also be made algorithmic. □

Given Lemmas 2.3 and 2.4, our desired approximate max-flow min-cut theorem follows immediately.

Theorem 2.6 ([5]). *Let f^* be the value of a maximum uniform concurrent multi-commodity flow problem on an n-node graph G with capacities $\{c_e\}$ and let R^* be the minimum cut in G. Then there exists a constant $c > 0$ such that $R^*/(c \cdot \log n) \leq f^* \leq R^*$. Moreover, a cut with sparsity at most $c \cdot f^* \log n$ can be found in polynomial time.*

Remark: Note that the proofs above are completely algorithmic, and we have seen all ingredients needed to *output* a cut with sparsity $O(f^* \log n)$ (and not just prove its existence). The only missing ingredient to make this into a polytime algorithm is a polytime solution for the dual LP, which has exponentially many constraints. This can be done in polynomial time using the Ellipsoid Method, which we will unfortunately not have time to cover in this course.

3 Applications

Here we outline two simple applications of Theorem 2.6 to approximation algorithms for some NP-hard problems. This theorem and its extensions to non-uniform demands, directed graphs and others have found applications to multiple other problems not covered in these notes. These include approximation algorithms for as disparate problems as crossing number, VLSI layout, minimum feedback arc set, among many others. See Leighton and Rao [5] for more.

3.1 Sparsest Cut

The first application is an immediate one. The *sparsest cut* problem is the problem of computing a cut $(U, \bar{U})$ of minimum density, defined to be $\frac{|(U,\bar{U})|}{|U||\bar{U}|}$, where $|(U, \bar{U})|$ denotes the set of edges crossing the cut. This problem can be approximated within an $O(\log n)$ term by applying Theorem 2.6 on the graph with unit capacities and unit demands. No approximation algorithm was known for this problem prior to Leighton and Rao's work. Moreover, this bound was the best known until the 2004 work of Arora et al. [1], who showed how to obtain an $O(\sqrt{\log n})$-approximation using an SDP relaxation of this problem and much more elaborate machinery, including expander flows.

3.2 Flux, Expansion and Minimum Quotient Separators

A second application of Theorem 2.6 is to the *edge expansion* or *flux* of a graph, defined to be $\min_{U \subseteq V} \frac{|(U,\bar{U})|}{\min\{|U|,|\bar{U}|}$. As $\max\{|U|, |\bar{U}|\} \in [n/2, n]$ always, the edge expansion of a cut is always within a factor of at most n from it sparsity. Consequently, the $O(\log n)$ approximation to sparsest cut of Section 3.1 also yields an $O(\log n)$ approximation to the cut of minimum edge expansion, also termed the *minimum quotient separator*.

References

[1] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 222–231, 2004.

[2] George Dantzig and Delbert R. Fulkerson. On the max-flow min-cut theorem of networks. In Harold W. Kuhn and Albert W. Tucker, editors, *Linear inequalities and related systems*, chapter 12, pages 215–223. Princeton university press, 1956.

[3] Peter Elias, Amiel Feinstein, and Claude Shannon. A note on the maximum flow through a network. *IRE Transactions on Information Theory*, 2(4):117–119, 1956.

[4] Lester R. Ford and Delbert R. Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8(3):399–404, 1956.

[5] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832, 1999.