

# Approximate Max-Flow Min-Cut Theorems (Course Notes Extension for COMP5703)

Shanshan Wang  
swang141@uottawa.ca

November 17, 2014

## 1 Introduction

In this report, we discuss two approximate max-flow min-cut theorems that first introduced by Tom Leighton and Satish Rao in 1988 [9] and extended in 1999 [10] for uniform multicommodity flow problems. In the theorems they first showed that for any  $n$ -node multicommodity flow problem with uniform demands, the max-flow is a  $\Omega(\log n)$  factor smaller than the min-cut; further, they proved the max-flow is always within a  $\theta(\log n)$ -factor of the cut. These two theorems also have substantial applications to the field of approximation algorithms, we briefly discuss two basic examples in chapter 3. Most of the content of this report are referencing [10] [9] [12] [13] [15] [1] [2] and [6].

### 1.1 Single Commodity Flow Problems

**Commodity** In a network flow problem, given a graph  $G = (V, E)$ , a source vertex  $s$  and a sink vertex  $t$ , where  $s, t \in E$ , then we call the vertex pair  $(s, t)$  a *commodity*.

In a single commodity flow problem, there is a graph  $G = (V, E)$  with each given a capacity, one of the nodes is designated as source and one as sink as an example shown in Figure 1. The objective is to route as much flow as possible from the source to the sink without violating the capacity of any edge. With the famous max-flow min-cut theorem introduced by Ford and Fulkerson[3], we know the value of max-flow equals the capacity of min-cut in a single commodity flow problem. Another point worth noting is that the min-cut is always an upper bound of the max-flow in a single commodity flow problem, because for any  $U \subseteq V$  that contains the source but not the sink, all flow from the source to the sink must be routed through edges in  $\langle U, \bar{U} \rangle$ , hence the total flow is limited by the capacity in the min-cut.

### 1.2 Muticommodity Flow Problems

Multicommodity flow problem is a network flow problem with multiple commodities between different source and sink nodes. There are  $K \geq 1$  commodities, each with source

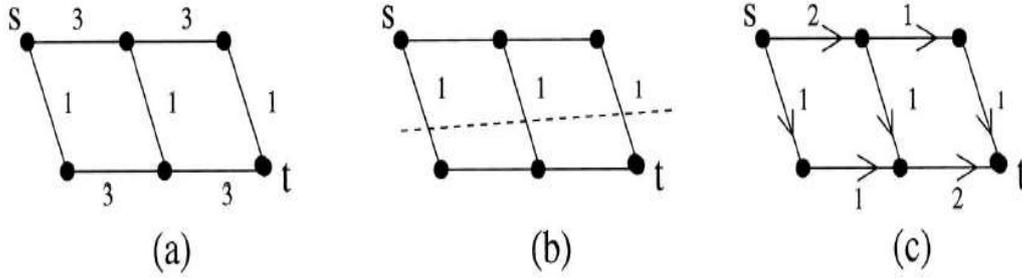


Figure 1: Single commodity flow problem (a) for which the min-cut (b) and the max-flow (c) are both 3.(Source from [10])

$s_i$ , sink  $t_i$  and demand  $D_i$ , as shown in Figure 2. The objective is to simultaneously route  $D_i$  units of commodity  $i$  from  $s_i$  to  $t_i$  for each  $i$  so that the total amount of all commodities passing through any edge is no greater than its capacity.

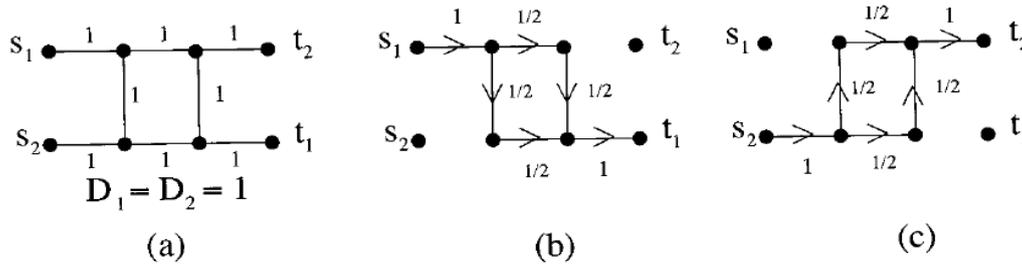


Figure 2: Solution to a 2-commodity flow problem (a) in which all edge capacities are 1. The routing of the first commodity is shown in (b) and the second commodity is shown in (c).(Source from [10])

**Definition 1.1.** [4] *Multicommodity flow problem:* Given a flow network  $G = (V, E)$ , where edge  $(u, v) \in E$  has capacity  $c(u, v)$ . There are  $k$  commodities  $K_1, K_2, \dots, K_k$ , defined by  $K_i = (s_i, t_i, D_i)$ , where  $s_i$  and  $t_i$  is the source and sink of commodity  $i$ , and  $D_i$  is the demand. The flow of commodity  $i$  along edge  $(u, v)$  is  $f_i(u, v)$ . Find an assignment of flow which satisfies the constraints:

- Capacity Constraints:  $\sum_{1 \leq i \leq k} f_i(u, v) \leq c(u, v)$ ,
- Flow Conservation:  $\sum_{w \in V} f_i(u, w) = 0$  when  $u \neq s_i, t_i$ ,
- Demand Satisfaction:  $\sum_{w \in V} f_i(s_i, w) = \sum_{w \in V} f_i(w, t_i) = D_i$ .

**Definition 1.2.** [10] *Max-flow in a multicommodity flow problem is the maximum value of  $f$  such that  $fD_i$  units of commodity  $i$  can be simultaneously routed for each  $i$  without violating any capacity constraints, where  $f$  is the common fraction of each commodity that is routed.*

**Definition 1.3.** [10] *Min-cut in a multicommodity flow problem is the minimum of all cuts of the ratio of the capacity of the cut to the demand of the cut, denoted as:*

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

where  $C(U, \bar{U})$  is the sum of capacities of the edges linking  $U$  to  $\bar{U}$ , and  $D(U, \bar{U})$  is the sum of the demands whose source and sink are on opposite sides of the cut that separates  $U$  from  $\bar{U}$ .

Similarly as for single commodity flow problem, max-flow is *always* upper bounded by the min-cut in a multicommodity flow problem. However, the problem of producing an integer flow satisfying all demands is NP-complete, even for only two commodities and unit capacities. If fractional flow are allowed, the problem can be solved in polynomial time through linear programming or through fully polynomial time approximation schemes.[4]

### 1.3 Uniform Muticommodity Flow Problems

In a uniform multicommodity flow problem, there is a commodity for every pair of nodes and the demand for every commodity is the same. Without loss of generality, the demand for every commodity is set to one. [9]. Since the relevant work in uniform multicommodity flow problems is very useful to solve NP-hard and NP-complete problems, especially the min-cut can provide good measures of the number of edges or the total weight of edges to partition a network into pieces of various sizes, it has been widely researched in many papers such as [13] [14] [12] and [15].

Based on the definition of the min-cut for multicommodity flow problems in Definition 1.3, for a uniform multicommodity flow problem, the demand across a cut  $\langle U, \bar{U} \rangle$  is simply the product of the number of nodes in  $U$  and the number of nodes in  $\bar{U}$ . In other words,

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

Hence, the min-cut of a uniform flow problem is

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

where  $C(U, \bar{U}) = \sum_{e \in \langle U, \bar{U} \rangle} C(e)$ . In the case when all capacities are 1, the min-cut is simply

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

## 2 Approximate Max-Flow Min-Cut Theorems

In this chapter, we introduce the two important theorems from [10][9]. Theorem 2.1 showed that for any  $n$ -node multicommodity flow problem with uniform demands, the

max-flow for the problem is a  $\Omega(\log n)$  factor smaller than the min-cut; furtherly, Theorem 2.2 showed that the max-flow is always within a  $\theta(\log n)$ -factor of the cut.

**Theorem 2.1.** [10] *For any  $n$ , there is an  $n$ -node uniform multicommodity flow problem with max-flow  $f$  and min-cut  $\psi$  for which  $f \leq O(\frac{\psi}{\log n})$ .*

*Proof.* We illustrate the proof by an example graph in Figure 3.

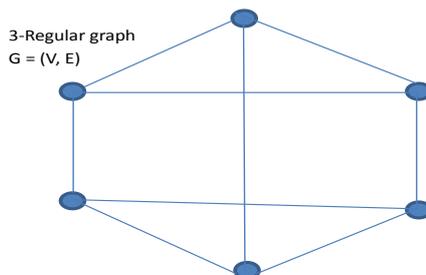


Figure 3: 3-regular graph  $G=(V,E)$

Let  $G = (V, E)$  be a 3-regular  $n$ -node graph (in the example  $n = 6$ ) with unit edge capacities for which

$$|\langle U, \bar{U} \rangle| \geq c \min \{|U|, |\bar{U}|\}$$

for some constant  $c > 0$  and all  $U \subseteq V$ . By the definition of  $G$ , we know that the min-cut is

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

Since  $G$  is 3-regular, there are at most  $n/2$  nodes within distance  $\log n - 3$  of any particular node  $v \in V$ . Hence, for at least half of the  $\binom{n}{2}$  commodities, the shortest path connecting the source and sink in  $G$  has at least  $\log n - 2$  edges. In order to sustain a flow of  $f$  for such a commodity, at least  $f(\log n - 2)$  capacity must be used by the commodity. Thus, in order to sustain a flow of  $f$  for all  $\binom{n}{2}$  commodities, the capacity in the network must be at least

$$(1/2) \binom{n}{2} f(\log n - 2).$$

Since the graph is 3-regular and has unit capacity edges, the total capacity is at most  $3n/2$ . Hence,

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

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

In other words, the max-flow for the uniform multicommodity flow problem in  $G$  is at least a  $\theta(\log n)$ -factor smaller than the min-cut. [10] [5] [8].  $\square$

**Ratio cost** Given a graph  $G = (V, E)$  and a cut  $\langle U, \bar{U} \rangle$  of  $G$ ,  $C(U, \bar{U})$  is the total capacities of all the edges in the cut. Then we refer to the quantity

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

as *ratio cost*.

**Dual of multicommodity flow problem** In general, the dual of multicommodity flow problem for a graph  $G$  is the problem of apportioning a fixed amount of weight (where weights are thought of as distances) to the edges of  $G$  so as to maximize the cumulative distance between the source and sink pairs. Alternatively, the dual can be thought of as apportioning the smallest amount of total distance so that the cumulative distances between the source and sink pairs is not too small. [10] More precisely, the dual of a  $k$ -commodity flow problem consists of finding a nonnegative distance  $d(e)$  for each edge  $e \in E$  so that

$$\sum_{1 \leq i \leq k} D_i d(s_i, t_i) \geq 1$$

and

$$\sum_{e \in E} C(e)d(e)$$

is minimized, where  $d(s_i, t_i)$  is the distance between the source and sink for the  $i$ th commodity in  $G$  with respect to the distance function. In the case of uniform multicommodity flow problem, the distance constraint is simply

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

we also let  $W$  to denote the total weight of the distance function, i.e.

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

**Theorem 2.2.** [10] For any uniform multicommodity flow problem,

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

where  $f$  is the max-flow and  $\psi$  is the min-cut of the uniform multicommodity flow problem.

To prove this theorem, we need to prove the properties specified in Lemma 2.3, 2.4 and 2.5 first, then by proving Lemma 2.6 as the last step we obtain the fact stated in Theorem 2.2. Basically, the general idea is as follows:

\* To get the max-flow, we derive it from the duality theory of LP, since an optimal distance function results in a *total weight* that is equal to the max-flow of the uniform multicommodity flow problem.

\* For the min-cut, the process is a little more complex, we need to follow a 3-stage algorithm to get it: Stage 1: consider the dual of uniform multicommodity flow problem and use the optimal solution to define a graph with distance labels on the edges; Stage 2: start from a source or a sink to grow a region in the graph until find a cut of small enough capacity separating the root from its mate; Stage 3: remove the growed region and repeat the process.

**Lemma 2.3.** [10] For any graph  $G$  with arbitrary edge capacities, any  $r > 0$ , and any distance function with total weight  $W$  it is possible to partition  $G$  into components with radius at most  $r$  so that the capacity of the edges connecting nodes in different components is at most  $4W \log n/r$ .

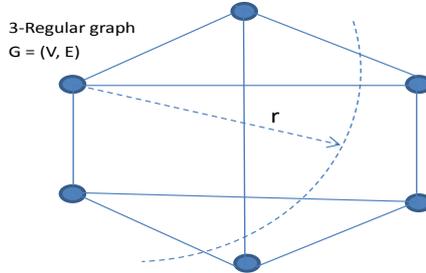


Figure 4: 3-regular graph  $G=(V,E)$  with radius  $r$  from a random vertex

*Proof.* Let  $C = \sum_{e \in E} C(e)$  denote the total capacity on the edges of  $G$ . When  $r \leq (4W \log n/C)$ , we can use the partition that leaves every node in a different component. Each such component will have  $r \geq 0$  and the capacity of the edges running between different components is at most  $C \leq (4W \log n/r)$ , then we are done.

If  $r > (4W \log n/C)$ , then we construct a second graph  $G'$  from  $G$  by replacing each edge  $e$  of  $G$  with a path of  $\lceil Cd(e)/W \rceil$  edges. Each edge along the path is assigned

capacity  $C(e)$  and distance 1. Then we show how to partition  $G$  by forming components in  $G'$ .

The components of  $G'$  are formed as follows: We begin by selecting an arbitrary node  $v \in G'$  that corresponds to a node in  $G$ . For each  $i \geq 0$ , define  $G'_i$  to be the subgraph of  $G'$  consisting of nodes and edges within distance  $i$  of  $v$ . (For  $G'$ , the distance between two nodes is defined to be the number of edges that are traversed in the shortest path connecting the nodes.) Let  $C_0 = (2C/n)$ , and for  $i > 0$ , define  $C_i$  to be the total capacity of the edges in  $G'_i$ . Let  $j$  denote the smallest value of  $i \geq 0$  for which  $C_i + 1 < (1 + \epsilon)C_i$  where  $\epsilon = (W \log n/rC) < 1/4$ . (There must be such  $j$  since for large enough  $i$ ,  $G'_{i+1} = G'_i$ ).

The nodes and edges in  $G'_j$  form the first component of the partition. The remaining components are found by removing  $G'_j$  from  $G'$  and then repeating the entire process. The process is repeated until there are no longer any nodes  $v \in G'$  that correspond to nodes in  $G$ .

Let  $C'$  denote the total initial capacity of  $G'$ . From the construction of  $G'$ , we know that

$$\begin{aligned} C' &= \sum_{e \in E} C(e) \lceil \frac{Cd(e)}{W} \rceil \\ &\leq \sum_{e \in E} C(e) + \frac{C}{W} \sum_{e \in E} C(e)d(e) \\ &= 2C \end{aligned}$$

From the method by which the components were formed, we know that the capacity of the edges leaving any component is at most  $\epsilon C_j$ , where  $C_j = (2C/n)$  if the component consists of a single node and  $C_j$  is the sum of the capacities on the edges contained in  $G'_j$ . Since the components of  $G'_j$  are disjoint, this means that the total capacity on all edges leaving all components in  $G'$  is at most

$$\epsilon(C' + nC_0) \leq 2\epsilon C' + 2\epsilon C = 4\epsilon C$$

The partition for  $G$  is derived from the components of  $G'$  in the natural way. In particular, two nodes of  $G$  are placed in the same component for  $G$  if and only if they were in the same component for  $G'$ . From the construction of the components, we know that any edge  $e \in G$  that links two components in  $G$  must correspond to a path of capacity  $C(e)$  edges in  $G'$  that was cut to form at least one of the corresponding components in  $G'$ . Hence, the total capacity of the edges linking different components in  $G$  is at most  $4\epsilon C = (4W \log n/r)$ .

Now it remains only to show that each component has small radius. This can be verified by considering a component with radius  $j$  in  $G'$ . Provided that  $j > 0$ , this component must have total capacity at least  $(1 + \epsilon)^j \frac{2C}{n} \leq 2C$  and thus (since  $\epsilon < 1/4$ ) that

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

Given a path of length  $l$  in  $G'$ , the corresponding path in  $G$  has length at most  $Wl/C$ . Hence, the radius of each component in  $G$  is at most

$$\frac{W \log n}{C\epsilon} = r$$

□

**Lemma 2.4.** [10] *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.* We apply the result of Lemma 2.3 with  $r = 1/2n^2$ . If one of the components formed during the construction of Lemma 2.3 contains at least  $2/3$  of the nodes of  $G$ , then we are done. Otherwise, we can divide the components into two sets so that each set contains at least  $n/3$  nodes. This division forms a cut with edge capacity at most

$$4W \log n/r = 8Wn^2 \log n$$

Since both sides of the cut have at least  $n/3$  nodes, the ratio cost of this cut is at most

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

□

**Lemma 2.5.** [10] *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*

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

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

*Proof.* The proof uses many of the same arguments that were used to prove Lemma 2.3. In particular, we start by defining the graph  $G'_i$  to be the subgraph of  $G'$  consisting of all nodes and edges that are within distance  $i$  of a node in  $T$ . (Recall that distance in  $G'$  is measured in terms of the number of edges that are traversed since every edge has distance 1 to  $G'$ .) We also define  $V_i$  to be the set of nodes of  $G$  that correspond to nodes in  $G'_i$ ,  $n_i = |V - V_i|$ ,  $R_i$  to be the ratio cost of the cut  $(V_i, V - V_i)$  in  $G$ , and  $R = \min \{R_i\}$ . Since  $n_i$  nodes of  $G$  are at distance at least  $i + 1$  from  $T$  in  $G'$  for all  $i$ , we know that

$$\sum_{u \in V-T} d_{G'}(T, u) = \sum_{i \geq 0} n_i$$

By the construction of  $G'$ , we also know that  $d_{G'}(T, u) \geq (C/W)d_G(T, u)$ . Hence, we can conclude that

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

Since  $|T| \geq 2n/3$ , the capacity in the cut  $\langle V_i, V - V_i \rangle$  of  $G$  is at least  $R_i n_i (2n/3) \geq (2nR_i n_i/3)$ . Hence, the capacity of the corresponding cut for  $G'_i$  in  $G'$  is also at least this amount. Since the total capacity in  $G'$  is at most  $C' \leq 2C$  (from the proof of Lemma 2.3), this means that

$$\sum_{i \geq 0} \frac{2nR_i n_i}{3} \leq 2C$$

and thus

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

Together with the lower bound derived, now we have

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

□

**Lemma 2.6.** [10] *Give a graph  $G$  and a distance function with total weight  $W$  that satisfies the distance constraint, we can find a cut with ratio cost  $O(W \log n)$ .*

*Proof.* We begin by partitioning the graph as in Lemma 2.3 with  $r = 1/2n^2$ . By Lemma 2.4, we can then either find a cut with ratio cost  $O(W \log n)$  (in which case, we are done) or we can find a component  $T$  with radius  $1/2n^2$  that contains at least  $2n/3$  nodes. In the latter case, we can apply Lemma 2.5 to find a cut with ratio cost  $O(W)$  (thereby concluding the proof), but we must first show that

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

The proof makes use of the fact that for any pair of nodes  $u, v \in V$

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

since  $T$  has radius  $1/2n^2$ . In particular,

$$\begin{aligned} \sum_{\{u,v\}} d(u, v) &= \frac{1}{2} \sum_{(u,v)} d(u, v) \\ &\leq \frac{1}{2} \sum_{(u,v)} (d(T, u) + d(T, v) + \frac{1}{n^2}) \end{aligned}$$

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

Because of the distance constraint, this means that

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

□

The proof of Theorem 2.2 follows from Lemma 2.6 and the fact that  $W = f$ . It is worth noting that each of the proofs in this section can be adapted to yield polynomial time algorithms for finding cuts and regions with low diameter.

### 3 Applications

#### 3.1 Sparsest Cut

**Sparsest Cuts** The *sparsest cut* of a graph  $G = (V, E)$  is a partition  $\langle U, \bar{U} \rangle$  for which

$$\frac{|\langle U, \bar{U} \rangle|}{|U||\bar{U}|}$$

is minimized where  $|\langle U, \bar{U} \rangle|$  denotes the number of edges connecting  $U$  to  $V - U$ .

Computing the sparsest cut of a graph is NP-hard.[11] The sparsest cut can be approximated to within an  $O(\log n)$  factor using the introduced algorithm. In this case, we can simply set all demands and capacities to be 1 and find a cut with ratio cost  $O(f \log n)$ .

#### 3.2 Flux, Expansion, and Minimum Quotient Separators

**Minimum Edge Expansion or Flux** Defined by

$$\alpha = \min_{U \in V} \frac{C(U, \bar{U})}{\min(|U|, |\bar{U}|)}$$

In other words, a graph has flux at least  $\alpha$  if every subset  $U$  with at most half of the nodes is connected to the rest of the graph with edges of total weight at least  $\alpha|U|$ .

**Minimum Quotient Separators** A cut that achieves the *flux* is called the *minimum quotient separator* and is related by a constant factor to the sparsest cut since

$$(n/2)\psi \leq \alpha \leq n\psi$$

for any  $n$ -node graph (because  $n/2 \leq \max(|U|, |\bar{U}|) \leq n$ ).

Computing the minimum quotient separator is NP-hard, even for unweighted graphs.[7] The mentioned algorithms for sparsest cut provide  $O(\log n)$  approximation algorithms for the flux and minimum quotient separator problems, even in the case where edges and nodes are weighted and where edges are directed. [10]

## 4 Exercises

**4.1.** *In a directed multicommodity flow problem, each edge has a specified direction, and flow is restricted to move only in the direction of each edge. Further, in a directed uniform multicommodity flow problem, we assume that the demand from  $u$  to  $v$  is 1 for each  $u \neq v$ . In this context, prove the following theorem based on the proof methodologies of Theorem 2 and the Lemmas: For any directed uniform multicommodity flow problem with  $n$  nodes,*

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

where  $f$  is the max-flow and  $\psi$  is the min-cut of the uniform multicommodity flow problem.

**4.2.** *In some applications, we desire to find a small cut in a graph  $G = (V, E)$  that partitions the graph into nearly equal-size pieces. In general, we say that a cut  $C < U, \bar{U} >$  is  $b$ -balanced or a  $(b, 1 - b)$ -separator for  $(b \leq 1/2)$  if*

$$bt(V) \leq t(U) \leq (1 - b)t(V)$$

where  $t(U)$  denotes the sum of the node weights in  $U$ . Finding  $b$ -balanced cuts of minimum edge weight is NP-hard, even in the special case when all node and edge weights are 1. Try to design an approximation algorithm for this problem, starting by finding an approximate sparsest cut.

## References

- [1] [http://en.wikipedia.org/wiki/Approximation\\_algorithm](http://en.wikipedia.org/wiki/Approximation_algorithm). [Online; accessed on 09-November-2014].
- [2] [http://en.wikipedia.org/wiki/Distance\\_\(graph\\_theory\)](http://en.wikipedia.org/wiki/Distance_(graph_theory)). [Online; accessed on 09-November-2014].
- [3] [http://en.wikipedia.org/wiki/Ford-Fulkerson\\_algorithm](http://en.wikipedia.org/wiki/Ford-Fulkerson_algorithm). [Online; accessed on 11-November-2014].
- [4] [http://en.wikipedia.org/wiki/Multi-commodity\\_flow\\_problem](http://en.wikipedia.org/wiki/Multi-commodity_flow_problem). [Online; accessed on 08-November-2014].
- [5] [http://en.wikipedia.org/wiki/Regular\\_graph](http://en.wikipedia.org/wiki/Regular_graph). [Online; accessed on 08-November-2014].
- [6] Sanjeev Khanna Chandra Chekuri and F.Bruce Shepherd. The all-or-nothing multicommodity flow problem. *Proc. of ACM STOC,2004*, 2004.
- [7] M.R. Garey and D.S. Johnson. Computers and intractability: A guide to the theory of np-completeness. *Freeman, San Francisco, Calif*, 1979.
- [8] Jon Kleinberg and Ronitt Rubinfeld. Short paths in expander graphs. *1996 IEEE Symposium on Foundations of Computer Science*, 1996.
- [9] Tom Leighton and Satish Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. *in Proc.29th IEEE Symposium on Foundations of Computer Science*, pages 422–431, 1988.
- [10] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J.ACM*, 46:787–832, 1999.
- [11] F. Matula, D.W.and Shahrokhi. The maximum concurrent flow problem and sparsest cuts. *Tech.Rep.Southern Methodist Univ.,Dallas,Tex*, 1986.
- [12] Vijay V.Vazirani Naveen Garg and Mihalis Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM J. COMPUT.*, 25:235–251, 1996.
- [13] R.Ravi Philip Klein, Ajit Agrawal and Satish Rao. Approximation through multicommodity flow. *in Proc.31st IEEE Symposium on Foundations of Computer Science*, pages 726–737, 1990.
- [14] R.Ravi Philip Klein, Ajit Agrawal and Satish Rao. Approximation through multicommodity flow. *Combinatorica*, 15:187–202, 1995.
- [15] S.Tragoudas. Improved approximations for the minimum-cut ratio and the flux. *Math.Systems Theory*, 29:157–167, 1996.