Lecture 9: Mar 6, 2015

Lecturer: Deeparnab Chakrabarty *Scribe: Saravanan K*

9.1 Cuts in Graphs

In this lecture we will discuss some of the problems related to cuts in graphs. Basically, given a graph $G(V, E)$, cuts form a subset $F \subset E$ such that, the graph $G \backslash F$ contains a set of disconnected partitions. Let us see three minimization problems related to cuts in graphs. They are

1. Minimum s-t cut problem
2. Multiway cut problem
3. Minimum Multicut problem

Notation: Given a graph $G(V, E)$ with edge costs $c_e \geq 0$, $\forall e \in E$. Then the cost of $F \subseteq E$ is the sum of costs of all edges in F. That is, cost of $F = \sum_{e \in F} c_e$.
First let us define the above problems. Then in the following sections we will discuss several approximation algorithms for these problems.

1. Minimum s-t cut problem: Given a graph $G(V, E)$ defined with costs $c_e \geq 0$ for all $e \in E$, with two fixed vertices $s, t \in V$. Then the minimum s-t cut is a subset $F \in E$ of minimum cost such that, s and t are disconnected in $G \backslash F$.

It is known that the maximum flow from s to t, which equals the minimum s, t cut, can be solved in polynomial time [1]. Here, we will use another argument (using Randomized Rounding) to prove that the minimum $s - t$ cut problem has a polynomial time 1-factor approximation algorithm, or in other words, an exact algorithm.

2. Multiway cut problem: Given a graph $G(V, E)$ defined with capacities $c_e \geq 0$ for all $e \in E$, and a set of vertices $\{s_1, s_2, \cdots, s_k\} \subseteq V$. The multiway cut is a subset $F \in E$ of minimum cost such that, in $G \backslash F$, s_i and s_j are disconnected, for all $1 \leq i, j \leq k$.

On careful observation, we note that when $k = 2$, multiway cut problem generalizes the minimum s-t cut problem. It has been proved [2] that when k is fixed (i.e., for constant k), this problem can be solved in polynomial time for planar graphs. However when k is not fixed, the multiway cut problem is known to be APX-$HARD$.

3. Minimum Multicut problem: Given a graph $G(V, E)$ defined with capacities $c_e \geq 0$ for all $e \in E$, with a set of pairs $\{(s_1, t_1), (s_2, t_2), \cdots, (s_k, t_k)\}$. Then the minimum multicut is a subset $F \subseteq E$ of minimum cost such that, in $G \backslash F$, there exists no path from s_i to t_i for all $1 \leq i \leq k$.

When $k = 1$, minimum multicut problem becomes the minimum s-t cut problem. The minimum multicut problem is known to be APX-$HARD$[3].

Definitions :

1. **Class** APX **:** The set of all optimization problems that belongs to the class NP having constant factor approximation algorithms belongs to the class APX.

2. **Class** APX-$HARD$ **:** The set of all problems that have a $PTAS$ *reduction* from every problem in APX belongs to the class APX-$HARD$.

9.2 1 - factor approximation for minimum s-t cut :

9.2.1 Linear Program

Let us define the variables x_e, for all edges $e \in E$. Let P_{st} be the set of all paths from s to t. Then, we can write LP relaxation for the minimum s-t cut problem by

$$\begin{aligned} & min \sum_{e \in E} c_e \cdot x_e \\ \text{subject to} \quad & \sum_{e \in P} x_e \geq 1, \quad \forall P \in P_{st} \\ & X_e \geq 0 \end{aligned}$$

However, the set of all paths from s to t is exponential in size, implies that there are exponential number of constraints. Therefore we intend to define a new LP for this problem.

Let $d(u, v)$ be the length of the shortest path from u to v. Then we write a new LP relaxation for minimum s-t cut problem as

$$\begin{aligned} & \min \sum_{e \in E} c_e \cdot x_e \\ & d(u, v) \leq x_{u,v} \quad \forall (u, v) \in E && (*) \\ & d(u, w) \leq d(u, v) + d(v, w) \quad \forall u, v, w \in V && \text{(triangle inequality)} \\ & d(s, t) \geq 1 \end{aligned}$$

Also by definition, we can verify that for a fixed set of x_e's, the above three inequalities must always be true. Proving the equivalence of this linear program to the previous linear program is left as an exercise. In compact we write this new LP as

$$\begin{aligned} & \min \sum_{e \in E} c_e \cdot x_e \\ & d_x(s, t) \geq 1 \end{aligned}$$

From now on the above notation is consistently followed throughout the lecture.

9.2.2 Algorithm

After solving the LP, we perform the following steps to compute our F. The algorithm is as follows

1. Sample r from $(0, 1)$ (assume uniform random distribution).
2. Find $S = \{v | d(s, v) \leq r\}$.
3. Output $F = \delta(S)$, (where $\delta(S)$ is the set of all edges having exactly one end point in S).

Observation 2: *F is a valid cut (that is, $s \in S$ and $t \notin S$).*
This is because of the fact that $d(s, s) = 0$, $d(s, t) \geq 1$ and $0 < r < 1$.

Observation 1: *If x_e's are integral, then our algorithm gives the ideal output.*
Proof: In this case all the edges that have $x_e = 1$ belongs to the set F. Hence by the definition of our LP we observe that the cost of F is minimized. That is, $\min \sum_{e \in E} c_e \cdot x_e = \min \sum_{e: x_e = 1} c_e \cdot x_e = \min F$.

9.2.3 Analysis

The expectation of F is given by

$$\mathbf{E}[c(F)] = \sum c_e \Pr[e \in F] \tag{9.1}$$

For a fixed edge $e = (u, v)$ (without loss of generality we assume $d(s, u) \leq d(s, v)$, $u \neq s$ and $v \neq s$), the probability that e belongs to F is given by,

$$\begin{aligned}
\Pr[e \in F] &= \Pr_{r \sim \texttt{Unif}(0,1)}[u \in S, v \notin S] \\
&= \Pr_{r \sim \texttt{Unif}(0,1)}[d(s, u) \leq r < d(s, v)] && (\because S = \{v | d(s, v) \leq r\}) \\
&\leq \frac{d(s, v) - d(s, u)}{1} \\
&\leq d(u, v) && \text{using triangle inequality} \\
&\leq x_{uv} && \text{using (*)}
\end{aligned}$$

Using the above in (6.1) we get

$$\begin{aligned}
\mathbb{E}[c(F)] &\leq \sum_{e \in E} c_e \cdot x_e \\
&= LP \\
&\leq OPT
\end{aligned}$$

Hence we state: the minimum s-t cut problem can be solved using a 1-factor approximation algorithm (meaning that, a polynomial time algorithm).

9.3 2-factor approximation for multiway cut

9.3.1 Linear Program

Similar to the minimum s-t cut problem, here is the LP relaxation for the multiway cut problem.

$$\begin{aligned}
&\min \sum_{e \in E} c_e \cdot x_e \\
&d_x(s_i, s_j) \geq 1 \quad : \forall i \neq j
\end{aligned}$$

9.3.2 Algorithm

Here we will analyze two similar algorithms (ALG 1 and ALG 2). In the former, we will obtain a $(k-1)$-factor approximation. While in the later, we will achieve a 2-factor approximation. The algorithms are as follows,

1. In ALG 1 : Sample r from $(0, 1)$
   In ALG 2 : Sample r from $(0, 1/2)$
2. Find $S_i = \{v \mid d(s_i, v) \leq r\}$, for all $i \in [k]$
3. Output $F = \bigcup_{i=1}^{k-1} \delta(S_i)$

9.3.3 Analysis

ALG 1 : Here we sample r from $(0,1)$.
For a fixed edge e, the probability that e belongs to F is given by

$$\begin{aligned}
\Pr_r[e \in F] &= \Pr_r[\exists i \,|\, e \in \delta(S_i)] \\
&\leq \sum_{i=1}^{k-1} \Pr_r[e \in \delta(S_i)] \\
&\leq \sum_{i=1}^{k-1} \frac{x_e}{1} \\
&= (k-1) \cdot x_e
\end{aligned}$$

Hence the expectation of cost of F is,

$$\begin{aligned}
\mathbb{E}[c(F)] &= \sum_{e \in E} c_e \cdot Pr_r[e \in F] \\
&\leq \sum_{e \in E} c_e \cdot (k-1) \cdot x_e \\
&= (k-1) \sum_{e \in E} c_e \cdot x_e \\
&= (k-1) \cdot LP
\end{aligned}$$

Therefore, when r is chosen from $(0,1)$, we obtain a $(k-1)$ factor approximation for ALG 1.
Now we will show that, ALG 2 is a 2-factor approximation, where we choose uniformly r from $(0,1/2)$.

ALG 2 : Here we sample r from $(0,1/2)$. Here we claim that there exists no vertex v such that $v \in S_i$ and $v \in S_j$, for some $i \neq j$. Suppose if such a vertex exists, then from step 2 of algorithm, we get $d(s_i, v) \leq r$ and $d(s_j, v) \leq r$ implying $d(s_i, s_j) \leq d(s_i, v) + d(s_j, v) \leq 2r < 1$, which is a contradiction to LP (i.e., $d(s_i, s_j) \geq 1$). Therefore, for every vertex u, we let S_u be the unique set in which u could possibly belong (that is, it is the set defined by the unique terminal s_u with $d(u, s_u) < 1/2$.)
Fix an $e = (u, v)$. We want to upper bound the probability that e belongs to F. If $s_u = s_v = s$, then the only terminal which can separate u and v is s. Assuming, $d(s,u) \leq d(s,v)$, we get the $\Pr_r[e \in F] = \Pr_r[d(s,u) \leq r < d(s,v)] \leq 2d(u,v)$.

Now suppose $s_u \neq s_v$. The probability e is cut is now the probability $u \in S_u$ or $v \in S_v$. This is because we know $v \notin S_u$ (since s_u and s_v are different), and similarly $u \notin S_v$. So,

$$\begin{aligned}
\Pr_r[e \in F] = \quad & \Pr_r[u \in S_u \text{ or } v \in S_v] \\
\leq \quad & \Pr_r[u \in S_u] + \Pr_r[v \in S_v] \\
= \quad & \Pr_r[d(s_u, u) \leq r] + \Pr_r[d(s_v, v) \leq r] && \text{(using above claim)} \\
= \quad & \frac{(1/2) - d(s_u, u)}{1/2} + \frac{(1/2) - d(s_v, v)}{1/2} \\
= \quad & 2(1 - [d(s_u, u) + d(s_v, v)]) \\
\leq \quad & 2 \cdot d(u, v) && (\because \; d(s_u, u) + d(u, v) + d(v, s_v) \geq d(s_u, s_v) \geq 1) \\
\leq \quad & 2 \cdot x_e && \text{using } (*)
\end{aligned}$$

Now,

$$\begin{aligned}\mathbb{E}[c(F)] &\leq 2 \cdot \sum_{e \in E} c_e \cdot x_e \\ &\leq 2 \cdot LP \\ &\leq 2 \cdot OPT\end{aligned}$$

Therefore we conclude our analysis by stating : the above algorithm is a 2-factor approximation for the multiway cut problem.

9.4 $O(\log k)$ factor approximation for multicut

9.4.1 Linear Program

Similar to previous problems, LP relaxation for multicut problem is given by

$$\begin{aligned}&\min \sum_{e \in E} c_e \cdot x_e \\ &d_x(s_i, t_i) \geq 1 \quad : \forall i \in [k]\end{aligned}$$

9.4.2 Algorithm

1. Sample r from $(0, 1/2)$.
2. Sample a random permutation σ of $\{1, 2, \cdots, k\}$.
3. Find $S_i = \{v \mid d(s_i, v) \leq r\} \setminus \bigcup_{j <_\sigma i} S_j$.
4. Output $F = \bigcup_{i=1}^{k} \delta(S_i)$.

Here $j <_\sigma i$ means that j comes before i in the permutation σ. We note that while picking S_i's we remove all the vertices that has already been picked by S_j's (where $j <_\sigma i$). In other words, we say that there exists no edge $e \in F$ such that $e \in S_a$ and $e \in S_b$ (for some $a \neq b$). We note that thisis a multicut. To see this we need to show that no S_i contains an (s_j, t_j) pair. Suppose it did – then $d(s_i, s_j) \leq r < 1/2$ and $d(s_i, t_j) < 1/2$ implying $d(s_j, t_j) < 1$ contradicting the LP constraint.

9.4.3 Analysis

Fix an edge (u, v). Let us define

$$\begin{aligned}\alpha_i &= \min\{d(s_i, u)\, ,\, d(s_i, v)\} \\ \text{and } \beta_i &= \max\{d(s_i, u)\, ,\, d(s_i, v)\}\end{aligned}$$

Note that $\beta_i - \alpha_i \leq d(u,v)$ for all i. Now, the probability that $e \in F$ is given by

$$\begin{aligned}
\Pr_{r,\sigma}(u,v) \in F] &= \Pr_{r,\sigma}[\exists i \,|\, \text{exactly one of } u \text{ or } v \text{ is in } S_i] \\
&\leq \sum_{i=1}^{k} \Pr_{r,\sigma}[\text{exactly one of } u \text{ or } v \text{ is in } S_i] \\
&= \sum_{i=1}^{k} \Pr_{r,\sigma}[(\alpha_i \leq r \leq \beta_i) \text{ and } \forall j <_\sigma i : (r < \alpha_j)] \\
&\leq \sum_{i=1}^{k} \Pr_{r,\sigma}[(\alpha_i \leq r \leq \beta_i) \text{ and } \forall j <_\sigma i : (\alpha_i < \alpha_j)] && (\because \alpha_i \leq r) \\
&\leq \sum_{i=1}^{k} \Pr_{r}[(\alpha_i \leq r \leq \beta_i)] \cdot \Pr_{\sigma}[\forall j <_\sigma i : (\alpha_i < \alpha_j)] \\
&\leq \sum_{i=1}^{k} \frac{\beta_i - \alpha_i}{1/2} \cdot \frac{1}{i} \\
&\leq \sum_{i=1}^{k} \frac{2 \cdot d_{(u,v)}}{i} \\
&= 2 \cdot d_{(u,v)} \cdot H_k
\end{aligned}$$

Now,

$$\begin{aligned}
\mathbf{E}[c(F)] &\leq \min \sum_{e \in E} c_e \cdot Pr_{r,\sigma}[e \in F] \\
&\leq 2H_k \cdot \sum_{e \in E} c_e \cdot d_{u,v} \\
&\leq 2H_k \cdot \sum_{e \in E} c_e \cdot x_e && \textit{using } (*) \\
&= 2H_k \cdot LP \\
&\leq 2H_k \cdot OPT \\
&\leq O(\log k) \cdot OPT
\end{aligned}$$

Hence the above algorithm is a $O(\log k)$-factor approximation for the multicut problem.

Now, let us go back to the multiway cut problem, where we design a new algorithm for a new LP relaxation with an approximation factor of 1.5.

9.5 1.5 factor approximation for multiway cut

Recall the multiway cut problem where we minimize the cost of F such that in $G \backslash F$, s_i and s_j are disconnected for all $i, j \in [k]$. We have seen the below LP relaxation.

$$\begin{aligned}
&\min \sum_{e \in E} c_e \cdot x_e \\
&d_x(s_i, s_j) \geq 1 \quad : \forall i \neq j
\end{aligned}$$

For all vertices $v \in V$, let us define variables $v :\ X^v = (X_1^v, X_2^v, \cdots, X_k^v)$ such that, $\sum_{i=1}^{k} X_i^u = 1$ and

$X_i^u \geq 0$. Using these variables we write

$$\min \quad \frac{1}{2} \cdot \sum_{(u,v)\in E} c_{u,v} \sum_{i=1}^{k} |X_i^u - X_i^v|$$
$$\forall u \,:\, \sum_{i=1}^{k} X_i^u = 1$$
$$\forall i \in [k] \,:\, X_i^{S_i} = 1$$
$$\forall i \in [k], u \in V \,:\, X_i^u \geq 0$$

However the above is not a linear program, because of the non-linearity of absolute value functions. Hence we define variables y_i^{uv} , $\forall (u,v) \in E, i \in [k]$ and write a new LP relaxation as follows

$$\min \frac{1}{2} \sum_{(u,v)\in E} c_{u,v} \sum_{i=1}^{k} y_i^{uv} \leq OPT$$
$$\forall u, v \,:\, y_i^{uv} \geq X_i^u - X_i^v$$
$$\forall u, v \,:\, y_i^{uv} \geq X_i^v - X_i^u$$
$$\forall u \,:\, \sum_{i=1}^{k} X_i^u = 1$$
$$\forall i \in [k] \,:\, X_i^{S_i} = 1$$
$$\forall i \in [k], u \in V \,:\, X_i^u \geq 0$$

Claim: For each edge $e(u,v) \in E$ we can add $(k-2)$ new vertices (say $u_1, u_2, \cdots, u_{k-2}$) such that X^u and X^v differ in at most 2 coordinates.
Proof: For a fixed edge (u,v) let $X^u = (X_1^u, X_2^u, \cdots, X_k^u)$ and $X^v = (X_1^v, X_2^v, \cdots, X_k^v)$ be the coordinate vectors.

1. Find a coordinate (say X_m) such that $X_m^u \geq X_m^v$.
2. Find another coordinate (say X_n) such that $X_n^u + X_m^u - X_m^v \leq 1$.
3. Add a vertex (say u_1) between u and v such that, $X_m^{u_1} = X_m^v$; $X_n^{u_1} = X_n^u + X_m^u - X_m^v$ and $X_i^{u_1} = X_i^u$ for all other X_i's.

Now u and u_1 differ in at most two coordinates (they are X_m, X_n). Also u_1 and v differ in at most $(k-1)$ coordinates (other than X_m). On careful observation, we note that using the same steps (for $(u_1, v))$) we can add one more vertex u_2 (between u_1 and v) such that, u_2 and v differ in at most $(k-2)$ coordinates. By recursively proceeding, we find that by adding $(k-2)$ new vertices (between u and v), every pair of adjacent vertices differ by at most $[k-(k-2)] = 2$ coordinates. This suffices the proof of claim.
Henceforth in our analysis, we assume that for all edges $(u,v) \in E$, X_u and X_v differ in at most 2 coordinates.

9.5.1 Algorithm :

Now we design an algorithm to find the set F by using the solution obtained from the above LP. The algorithm is as follows :

1. Sample a random permutation σ from $\{1, 2, \cdots, k\}$.
2. Sample r randomly from (0,1].
3. Find $S_i := \{v \,|\, X_i^v \geq r\} \setminus \cup_{j <_\sigma i} S_j$, for all $i \in [k]$.
4. Output $F = \cup_{i=1}^{k} \delta(S_i)$

9.5.2 Analysis :

Let $e(u,v) \in E$ be an edge such that $X^u = (u_1, u_2, \cdots, u_k)$ and $X^v = (u_1 - \delta, u_2 + \delta, u_3, u_4, \cdots, u_k)$.
By our algorithm we say, for coordinates $3, 4, \cdots, k$, the sets $S_3, S_4, \cdots, S_k$ will either pick both u and v or pick none of the vertices in $\{u, v\}$. Therefore for the fixed edge $e(u,v)$, the probability that $(u,v) \in F$ is given by

$$\Pr_{r,\sigma}[(u,v) \in F] \leq \Pr_{r,\sigma}[S_1 \text{ cuts } (u,v)] + \Pr[S_2 \text{ cuts } (u,v)] \tag{9.2}$$

Till certain point, let us assume

$$u_2 + \delta \geq u_1 \qquad \textit{(Assumption 9.3)}$$

Case (i) Suppose $1 <_\sigma 2$:
If S_1 cuts (u,v) then $u_1 - \delta \leq r \leq u_1$ (by algorithm definition). Therefore,

$$\Pr_r[S_1 \text{ cuts } (u,v)] \leq \frac{X_u^1 - X_v^1}{1} = \delta$$

Also, S_2 cuts (u,v) only if neither of u of v are in S_1 and $u_2 \leq r \leq u_2 + \delta$. Therefore,

$$\begin{aligned}
\Pr_r[S_2 \text{ cuts } (u,v)] &\leq \Pr_r[(S_1 \text{ `spares' } u \text{ and } v \text{ and } S_2 \text{ cuts } (u,v)] \\
&= \Pr_r[r \geq u_1 \text{ and } u_2 \leq r \leq u_2 + \delta] \\
&\leq \Pr_r[u_2 + \delta \geq u_1 \text{ and } u_2 \leq r \leq u_2 + \delta] \\
&= \Pr_r[u_2 \leq r \leq u_2 + \delta] && \text{(by assumption 9.3)} \\
&\leq \delta
\end{aligned}$$

From (9.2) we get,

$$\Pr_r[\,(u,v) \in F \;\mid\; 1 <_\sigma 2\,] \leq \delta + \delta = 2\delta \tag{9.3}$$

Case (ii) Suppose $2 <_\sigma 1$:
Similar to **Case (i)**, we get

$$\begin{aligned}
\Pr_r[S_2 \text{ cuts } (u,v)] &= \Pr_r[u_2 \leq r \leq u_2 + \delta] \\
&= \frac{\delta}{1} = \delta
\end{aligned}$$

Similarly,

$$\begin{aligned}
\Pr_r[S_1 \text{ cuts } (u,v)] &= \Pr_r[S_1 \text{ not cutting } (u,v) \text{ and } S_2 \text{ cuts } (u,v)] \\
&= \Pr_r[r > u_2 + \delta \text{ and } u_1 - \delta \leq r \leq u_1] \\
&= 0 && \textit{(by (9.3), } u_2 + \delta < r \leq u_1 \textit{ never occurs)}
\end{aligned}$$

From (9.2) we get,

$$\Pr_r[(u,v) \in F \;\mid\; 2 <_\sigma 1] \leq \delta + 0 = \delta \tag{9.4}$$

On careful observation, we note that $\Pr_\sigma[1 <_\sigma 2] = 1/2$ and $\Pr_\sigma[1 <_\sigma 2] = 1/2$. Therefore under the assumption that $u_2 + \delta \geq u_1$ we get,

$$\begin{aligned}
\Pr_{r,\sigma}[(u,v) \in F] &= (1/2) \cdot (\Pr_{r,\sigma}[(u,v) \in F \mid (1 <_\sigma 2)] + \Pr_{r,\sigma}[(u,v) \in F \mid (2 <_\sigma 1)]) \\
&\leq (1/2) \cdot (2\delta + \delta) \\
\implies \Pr_{r,\sigma}[(u,v) \in F] &\leq \frac{3\delta}{2} \qquad (**)
\end{aligned}$$

The above probability is true when *assumption 9.3* is true (that is, $u_2 + \delta \geq u_1$). Next we analyze the case where we assume $u_2 + \delta < u_1$. On careful observation, we note that the same analysis works for the new assumption. This is because of the symmetric nature of the coordinates X_1 and X_2. Here we get,

$$\begin{aligned}
\Pr_r[(u,v) \in F \mid (1 <_\sigma 2)] &\leq \delta \\
\text{and } \Pr_r[(u,v) \in F \mid (1 <_\sigma 2)] &\leq 2\delta
\end{aligned}$$

Therefore under the assumption that $u_2 + \delta < u_1$, we get

$$\begin{aligned}
\Pr_{r,\sigma}[(u,v) \in F] &= (1/2) \cdot (\Pr[(u,v) \in F \mid (1 <_\sigma 2)] + \Pr[(u,v) \in F \mid (2 <_\sigma 1)]) \\
&\leq (1/2)\delta + 2\delta \\
\implies \Pr_{r,\sigma}[(u,v) \in F] &\leq \frac{3\delta}{2} \qquad (***)
\end{aligned}$$

From $(**)$ and $(***)$ we state: for a fixed edge $e(u,v)$, the probability that $e \in F$ is given by

$$\Pr_{r,\sigma}[(u,v) \in F] \leq \frac{3\delta}{2} \tag{9.5}$$

Hence, the expectation of the cost of F is

$$\begin{aligned}
\mathbf{E}[c(F)] &= \sum_{e(u,v)\in E} c_e \cdot \Pr_{r,\sigma}[e(u,v) \in F] \\
&\leq (3/2).(1/2) \sum_{e(u,v)\in E} c_e \cdot 2\delta && \textit{using (9.5)} \\
&\leq (3/2).(1/2) \sum_{e(u,v)\in E} c_e \cdot \sum_{i=1}^{k} (X_i^u - X_i^v) && (\because \text{only two coordinates differ, by at most } \delta) \\
\implies \mathbf{E}[c(F)] &\leq (3/2).cost[LP] \leq (3/2).OPT
\end{aligned}$$

Thus based on the above analysis, we observed that the algorithm provides a 1.5 - factor approximation for solving the multiway cut problem.

References

[1] Ford, L. R.; Fulkerson, D. R., Maximal flow through a network, *Canadian Journal of Mathematics 8:399-404*, 1956

[2] E. Dahlhaus, D.S Johnson, C.H. Papadimitriou,P.D. Seymour and M. Yannakakis, The Complexity of Multiterminal Cuts, *SIAM Journal on Computing*, 1994

[3] Erik D. Demaine, Dotan Emanuel, Amos Fiat and Nicole Immolica, Correlation Clustering in General Weighted Graphs, *Theoretical Computer Science*, 2006

[4] http://en.wikipedia.org/wiki/APX