Minimum s, t-cut and Multiway Cut[1]

- ***Cut Problems.*** In the next few lectures we look at various cut problems in graphs. The input will be an undirected graph $G = (V, E)$ with non-negative costs $c(e)$ on edges. The objective for each problem is to select a subset $F \subseteq E$ of these edges with minimum cost $c(F) := \sum_{e \in F} c(e)$, so that upon deleting F certain vertices get *cut* or *disconnected*.

- ***Minimum s, t-cut Problem and the Distance based LP.*** We begin with a problem which has an exact algorithm and which you have seen before in your undergraduate algorithms class. It is the min s, t-cut problem. The objective is to select F such that after deleting F, we disconnect s from t. However, we will look at an LP relaxation for the problem, and argue that it is *exact*. Let's begin with the linear program.

  We have variables x_e for every edge $e = (u, v)$ indicating whether we select (u, v) in our solution or not. The objective is clear, it is to minimize $\sum_{e \in E} c(e)x_e$. What about the set of constraints? We need that in *every* path from s to t, we select at least one edge into F; if not, then s and t would remain connected. We could write a collection of exponentially many constraints and indeed we could solve it using the ellipsoid method. However, we write a succinct LP. It stems from the following interpretation. If we think of x_e as the "length" of the edge e, then saying that every path contains at least one edge in F is equivalent to saying that the length of this path is at least 1. In other words, the constraint can be captured by saying that the "distance" from s to t induced by these lengths x_e has to be at least 1.

  How do we capture these distances? For *every* pair of nodes (not necessarily neighboring) we now introduce a variable d_{uv} indicating the distance. We need $d_{st} \geq 1$. How should the d-variables relate with the x-variables? Well, for any *edge* (u, v), the distance d_{uv} is at most the length x_{uv}. Finally, the fact that the d's induce a "distance", we introduce the "triangle inequality constraint" : between any triple of vertices $\{u, v, w\}$, we must have $d_{uw} \leq d_{uv} + d_{vw}$. Note that the true shortest path distances do satisfy this, and thus the LP below is a valid relaxation.

$$\mathsf{lp} := \min \sum_{e \in E} c(e)x_e \qquad (s, t\text{-min cut LP})$$

$$d_{uv} \leq x_e, \qquad \forall e \in E, e = (u, v) \qquad (1)$$

$$d_{uw} \leq d_{uv} + d_{vw}, \quad \forall i \in F,\ \forall \{u, v, w\} \subseteq V \qquad (2)$$

$$d_{vv} = 0, \qquad \forall v \in V \qquad (3)$$

$$d_{st} \geq 1 \qquad (4)$$

Exercise: ☕♡ *Write the dual for the LP above. Interpret the dual.*

[1] *Lecture notes by Deeparnab Chakrabarty. Last modified : 17th Feb, 2022*
These have not gone through scrutiny and may contain errors. If you find any, or have any other comments, please email me at deeparnab@dartmouth.edu. Highly appreciated!

- ***An Exact algorithm via Randomized Rounding.*** We now show a randomized algorithm which returns an s, t cut with probability 1 with expected cost $\leq \mathsf{lp}$. This should remind of another algorithm we saw in class earlier. Furthermore, it also shows randomization is completely unnecessary. Here is the algorithm.

1: **procedure** RANDOMIZED MIN s, t-CUT($G = (V, E)$, $c(e) \geq 0$ on edges):
2: Solve (s, t-min cut LP) to obtain x_e's and d_{uv}'s.
3: Randomly sample $r \in (0, 1)$ uniformly.
4: $S := \{v : d_{sv} \leq r\}$.
5: **return** $F := \partial S$.

Theorem 1. RANDOMIZED MIN s, t-CUT returns a set F whose removal disconnects s and t with probability 1, and $\mathbf{Exp}[\sum_{e \in F} c(e)] = \mathsf{lp}$.

Proof. First, let us observe that F is a valid min-cut with probability 1. Indeed, the set S contains s since $d_{ss} = 0$ and $t \notin S$ since $d_{st} \geq 1 > r$. Thus, ∂S disconnects s from t irrespective of r.

Now fix an edge $e := r(u, v)$ and let us analyze the probability $(u, v) \in F$. We perform this a bit carefully as similar calculations will be used at least twice more. Let $\mathbf{1}_{e \in F}$ be the event $e \in F$. We note that this event is the union of two events.

$$\mathbf{1}_{e \in F} = \mathbf{1}_{u \in S, v \notin S} \cup \mathbf{1}_{u \notin S, v \in S}$$

At this point, without loss of generality, let us assume $d_{su} \leq d_{sv}$ (otherwise swap their names). This allows us to infer that $\mathbf{1}_{u \notin S, v \in S}$ cannot occur: if $v \in S$, then $d_{sv} \leq r$ which would imply $d_{su} \leq r$. Therefore, the only event to analyze is $\mathbf{1}_{u \in S, v \notin S}$. Therefore,

$$\mathbf{Pr}[\mathbf{1}_{e \in F}] = \mathbf{Pr}[\mathbf{1}_{u \in S, v \notin S}] = \mathbf{Pr}[d_{su} \leq r < d_{sv}]$$

What is the probability that this random r is between d_{su} and d_{sv}? Well, triangle inequality (2) tells us that $d_{sv} \leq d_{su} + d_{uv}$, and (1) tells us $d_{sv} \leq d_{su} + x_e$. Thus the event $d_{su} \leq r < d_{sv}$ is a subset of the event $d_{su} \leq r < d_{su} + x_e$. Therefore, we get

$$\mathbf{Pr}[\mathbf{1}_{e \in F}] = \mathbf{Pr}[d_{su} \leq r < d_{sv}] \leq \mathbf{Pr}_{r}\left[r \in [d_{su}, d_{su} + x_e]\right]$$

And the final probability, the chance that a random $r \in [0, 1]$ lies in the interval $[d_{su}, d_{su} + x_e]$ is precisely $\min(x_e, 1 - d_{sv}) \leq x_e$. In sum, the probability a particular edge e lies in F is at most x_e. Applying linearity of expectation gives us $\mathbf{Exp}[\sum_{e \in F} c(e)] \leq \sum_{e \in E} c(e) x_e = \mathsf{lp}$. $\square$

Remark: *As in the case of vertex cover in bipartite graphs, the above shows that running the algorithm above with* ***any*** $r \in (0, 1)$ *would return a solution with cost exactly equal to* lp. *Do you see this?*

- ***Multiway Cut Problem.*** Let's move to an NP-hard problem. We are given k vertices $\{s_1, \ldots, s_k\}$. The objective now is to find F of minimum cost such that in $G \setminus F$ every s_i is disconnected from every other s_j. When $k = 2$, this is simply the minimum s, t-cut problem. Turns out, this problem is NP-hard even when $k = 3$.

  We begin with the LP very similar to (s, t-min cut LP). In fact, the only difference is that (4) is replaced by the natural generalization.

$$\mathsf{lp} := \min \sum_{e \in E} c(e) x_e \tag{Multiwaycut LP}$$

$$d \text{ satisfies (1),(2),(3)}$$

$$d_{s_i s_j} \geq 1, \qquad \forall i \neq j \tag{5}$$

- ***A 2-approximate algorithm via randomized rounding.*** The algorithm and analysis are similar to that of min-cut, but subtly different. First, the random radius r is selected uniformly at random from $(0, 1/2)$. Indeed, this leads to the factor 2. The algorithm is described below

```
1: procedure RANDOMIZED MULTIWAY CUT(G = (V, E), c(e) ≥ 0 on edges, s_1, ..., s_k):
2:     Solve (Multiwaycut LP) to obtain x_e's and d_uv's.
3:     Randomly sample r ∈ (0, 1/2) uniformly.
4:     For 1 ≤ i ≤ k, define S_i := {v : d_sv ≤ r}.
5:     return F := ⋃_{i=1}^k ∂S_i.
```

Theorem 2. RANDOMIZED MULTIWAY CUT returns a set F whose removal disconnects every s_i from every other s_j with probability 1, and and $\mathbf{Exp}[\sum_{e \in F} c(e)] = 2\mathsf{lp}$.

Proof. Once again, it should be clear that F is a valid multiway cut for any choice of $0 \leq r < 1/2$ (indeed, even $r < 1$ would lead to a valid solution). The interesting thing is the expected cost. Fix an edge $e := (u, v)$; we now prove that the probability $(u, v) \in F$ is at most $2x_e$.

We begin by making a key observation. For any vertex $v \in V$, there can be *at most* one value $1 \leq i \leq k$, call this $\phi(v)$, such that v can lie in $S_{\phi(v)}$. Put differently, v cannot lie in any other S_i for $i \neq \phi(v)$. It could be that for some r, v lies in none of the S_i's, but if it does, then that S_i is $S_{\phi(v)}$. The reason is simple. Suppose v could lie in S_i and S_j for $i \neq j$. Then $d(v, s_i) < 1/2$ as for some radius r we have $d(v, s_i) \leq r$. Similarly, $d(v, s_j) < 1/2$. But then triangle inequality would imply $d_{s_i s_j} < 1$, which would be a contradiction.

Now let's get back to the edge $e := (u, v)$. Say $\phi(u) = \phi(v) = i$. Then, the edge $(u, v) \in F$ if and only if $u \in S_i, v \notin S_i$, or vice-versa. This case is similar to the s, t-minimum cut argument; the only difference is that the radius is drawn in $[0, 1/2]$ and thus in the probability calculation, we have a $1/2$ in the denominator, which leads to the assertion: $\mathbf{Pr}[(u, v) \in F] \leq 2x_e$. We leave the details to the reader as an exercise.

Now suppose $\phi(u) = i$ and $\phi(v) = j$, and $i \neq j$. Notice that $(u, v) \in F$ if and only if $u \in S_i$ ***or*** $v \in S_j$; this is because if $u \in S_i$ we are sure $v \notin S_i$ (since $\phi(v) \neq i$). Therefore, we get

$$\mathbf{Pr}[e \in F] \;=\; \mathbf{Pr}[u \in S_i \;\textbf{ or }\; v \in S_j] \;\leq\; \mathbf{Pr}[u \in S_i] + \mathbf{Pr}[v \in S_j]$$

Next, note that $\mathbf{Pr}[u \in S_i] = \mathbf{Pr}[d(s_i, u) \leq r] \leq \frac{0.5 - d_{s_i u}}{0.5} = 1 - 2d_{s_i u}$, since $r \in [d_{s_i u}, 0.5]$ for the event to occur. Similarly, $\mathbf{Pr}[v \in S_j] \leq 1 - 2d_{s_j v}$. Adding them up, we get

$$\mathbf{Pr}[e \in F] \;\leq\; 2 \cdot \left(1 - d_{s_i u} - d_{s_j v}\right) \;\leq\; 2d_{uv} \leq 2x_e$$

where the middle inequality is obtained using triangle inequality and (5): $1 \leq d_{s_i s_j} \leq d_{s_i u} + d_{uv} + d_{v s_j}$, implying $1 - d_{s_i u} - d_{s_j v} \leq d_{uv}$. $\square$

Exercise: ☕ *Explain how you will modify the above algorithm to obtain an* $2\left(1 - \frac{1}{k}\right)$*-approximation.*

Exercise: ☕ *Prove the integrality gap of* (Multiwaycut LP) *is at least* $2\left(1 - \frac{1}{k}\right)$.

Notes

The $2(1 - 1/k)$-approximation and the NP-hardness of the MULTIWAY CUT problem is from the paper [4] by Dahlhaus, Johnson, Papadimitriou, Seymour, and Yannakakis. The presentation above for s, t-cut is probably folklore, but it forms a basis for the $\frac{3}{2}$-factor algorithm in the paper [3] by Calinescu, Karloff, and Rabani. This paper introduced a new LP-relaxation (as one has to given the exercise above) based on "embeddings" on a simplex. The integrality gap of this LP is still not fully understood, and in recent years, there has been a lot of active work on it. A notable result is in the paper [5] by Manokaran, Naor, Raghavendra and Schwartz where the authors prove that the integrality gap of this LP captures the UGC-hardness of multiway cut; if one obtains a better approximation factor than the integrality gap by some other means, one refutes the UGC. An elegant $\frac{4}{3}$-approximation is present in the paper [2] using a randomized rounding idea using exponential random variables. The current best upper bound on the integrality gao is 1.2965 from the paper [6] by Sharma and Vondrák, and the best lower bound is 1.20016 from the paper [1] by Bérczi, Chandrasekharan, Király, and Madan.

References

[1] K. Bérczi, K. Chandrasekaran, T. Király, and V. Madan. Improving the integrality gap for multiway cut. *Mathematical Programming*, 183(1):171–193, 2020.

[2] N. Buchbinder, J. Naor, and R. Schwartz. Simplex partitioning via exponential clocks and the multiway cut problem. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 535–544, 2013.

[3] G. Calinescu, H. Karloff, and Y. Rabani. An Improved Approximation Algorithm for Multiway Cut. *J. Comput. Syst. Sci.*, 60(3):564–574, 2000.

[4] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The Complexity of Multiterminal Cuts. *SIAM Journal on Computing (SICOMP)*, 23(4):864–894, 1994.

[5] R. Manokaran, J. Naor, P. Raghavendra, and R. Schwartz. Sdp gaps and ugc hardness for multiway cut, 0-extension, and metric labeling. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 11–20, 2008.

[6] A. Sharma and J. Vondrák. Multiway cut, pairwise realizable distributions, and descending thresholds. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 724–733, 2014.