Lecture 7

Lecturer: David P. Williamson *Scribe: Gwen Spencer*

1 Global min-cut in undirected graphs

In this lecture we'll take a few more steps pursuing the global min-cut problem. In particular, we'll look at two algorithms for the global min-cut problem on undirected graphs that are not flow-based.

Global min-cut in an undirected graph

- **Input:**
  - Undirected graph $G = (V, E)$
  - Arc capacities $u_{ij} > 0 \ \forall (i,j) \in E$
- **Goal:** Find $S \subset V, S \neq \emptyset$ that minimizes $u(\delta(S)) = \sum_{(i,j)\in\delta(S)} u_{ij}$.

We define $\delta(S)$ for $S \subset V$ as

Definition 1 $\delta(S) = \{(i,j) \in E : i \in S, j \notin S \text{ or } i \notin S, j \in S\}$

We also define $\delta(A, B)$ for two vertex sets $A, B \subset V$, $A \cap B = \emptyset$ as

Definition 2 $\delta(A,B) = \{(i,j) \in E : i \in A, j \in B \text{ or } i \in B, j \in A\}$

See Figure 1 for a pictorial explanation of $\delta(A, B)$.

1.1 MA Orderings

Let us now consider the following greedy algorithm:

MA (max adjacency) ordering

```
S ← {v_1}
For i ← 2 to n
    Choose v_i to maximize u(δ(S, {v})) ∀v ∈ V − S
    S ← S ∪ {v_i}
```

Given some arbitrarily chosen vertex v_1, the algorithm returns an ordering of the vertices. In each iteration, the algorithm looks at all vertices not in the set S and picks the one which maximizes the capacity of arcs connecting it to nodes in S. We will prove the following claim:

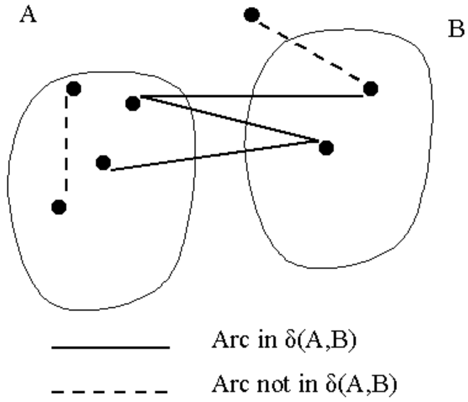

Figure 1: Example of $\delta(A, B)$.

Claim 1 *For MA ordering* $v_1, \ldots, v_n$, $\{v_n\}$ *is a minimum* v_{n-1}-v_n *cut (it has the minimum capacity of all cuts which separate* v_{n-1} *from* v_n*).*

We'll prove this claim in a moment. First we'll show that if it holds, we can use MA-ordering as a subroutine to compute the global min-cut. To see this, let S^* be a global min-cut. Consider 2 cases:

Case 1: $v_n \in S^*, v_{n-1} \notin S^*$ (or vice versa).

Case 2: $v_n, v_{n-1} \in S^*$ (or $v_n, v_{n-1} \notin S^*$).

If Case 1 happens then the fact that S^* is global min-cut implies that S^* is a minimum v_n-v_{n-1} cut, and since $\{v_n\}$ is also a minimum v_{n-1}-v_n cut we get that $u(\delta(S^*)) = u(\delta(\{v_n\}))$.

If Case 2 happens then we can "contract" v_n and v_{n-1} to a single vertex (we know they occur on the same side of the global min-cut).

The following algorithm uses MA orderings:

Global min-cut using MA orderings

$cutval \leftarrow \infty, S \leftarrow \emptyset$
While $|V| > 1$
    Compute MA ordering $v_1, v_2, ..., v_n$
    If $u(\delta(v_n)) < cutval$
        $cutval \leftarrow u(\delta(v_n))$, $S \leftarrow \{v_n\}$
        Contract v_{n-1} and v_n into a single node.
Return S.

In every iteration, either the algorithm finds a global min-cut or there is a contraction that decreases the size of the vertex set. Since the vertex set is finite, the number of contractions is limited: at most $n-1$ contractions can occur, so at most n iterations of the algorithm described produce the global mincut.

An example of the above algorithm is presented in Figure 2.

Now to prove claim 1. We'll need the following lemma.

Lemma 2 *Let $\lambda(G, s, t)$ denote value of the min s-t cut in G. Then for any three vertices $p, q, r \in V$, $\lambda(G, p, q) \geq \min(\lambda(G, r, q), \lambda(G, p, r))$.*

Proof: Let S be the min p-q cut of the graph and $p \in S$. Now suppose $r \in S$. Then, $\lambda(G, p, q) \geq \lambda(G, r, q)$, since S is also an r-q cut. If $r \notin S$, then $\lambda(G, p, q) \geq \lambda(G, p, r)$ since S is also a p-r cut. In either case, the result holds. $\square$

Proof of Claim 1: We know that by the definition of the min cut $\lambda(G, v_{n-1}, v_n) \leq u(\delta(v_n))$. We need to show that $\lambda(G, v_{n-1}, v_n) \geq u(\delta(v_n))$. We do this through an induction on the number of nodes and edges, $|E| + |V|$.

- The base case, i.e. when either $|E| = 0$ or $|V| = 2$, holds trivially.
- For the inductive case, there are two possibilities

  (i) $(v_{n-1}, v_n) \in E$:
  Let $(v_{n-1}, v_n) = e$, $G' \leftarrow G - e$, and let $\delta'(S)$ denote the capacity of the set of edges in G' that have exactly one vertex in S. Now, observe that $v_1, v_2, ..., v_n$ is still an MA ordering of G', and

$$\begin{aligned} u(\delta(v_n)) &= u(\delta'(v_n)) + u_e \\ &= \lambda(G', v_{n-1}, v_n) + u_e \\ &= \lambda(G, v_{n-1}, v_n). \end{aligned}$$

  The second equality is from induction and the final equality is because a v_{n-1}-v_n cut in G' has the same value in G (except the capacity of edge e is added, since in G edge e is in any v_{n-1}-v_n cut).

  (ii) $(v_{n-1}, v_n) \notin E$:
  In this case, we need to apply the inductive hypothesis twice. First, let $G' \leftarrow G - v_{n-1}$. Note that $v_1, v_2, \ldots, v_{n-2}, v_n$ is an MA ordering in G', and by the inductive hypothesis,

$$\begin{aligned} u(\delta(v_n)) &= u(\delta'(v_n)) \\ &= \lambda(G', v_{n-2}, v_n) \\ &\leq \lambda(G, v_{n-2}, v_n). \end{aligned}$$

  The last inequality follows since the cut in G separating v_{n-2} and v_n has no greater value in G' (the edges of G' are a subset of the edges of G).
  Now, let $G'' \leftarrow G - v_n$. Again, $v_1, v_2, \ldots, v_{n-1}$ is an MA ordering in G'', and by the construction of the ordering, and the inductive hypothesis,

$$\begin{aligned} u(\delta(v_n)) &\leq u(\delta(v_{n-1})) \\ &= u(\delta''(v_{n-1})) \\ &= \lambda(G'', v_{n-2}, v_{n-1}) \\ &\leq \lambda(G, v_{n-2}, v_{n-1}). \end{aligned}$$

The first inequality follows by the choice of v_{n-1} over v_n in the MA ordering for G, plus the assumption that there is no (v_{n-1}, v_n) edge in G. The last inequality follows since the cut in G separating v_{n-2} and v_{n-1} has no greater value in G'' than it does in G.

Now using Lemma 2,

$$\lambda(G, v_{n-1}, v_n) \geq \min(\lambda(G, v_{n-2}, v_{n-1}), \lambda(G, v_{n-2}, v_n)) \geq u(\delta(v_n)).$$

Therefore by the principle of mathematical induction, $\lambda(G, v_{n-1}, v_n) \geq u(\delta(v_n))$ holds for any number of vertices and edges. This proves the claim.

□

The basic idea of the MA ordering is due to Nagamochi and Ibaraki (1992). Simplifications were made by Frank (1994) and Stoer and Wagner (1997). Using Fibonacci heaps, an MA ordering of a graph can be computed in $O(m + n \log n)$ time. Thus, the algorithm to compute a global min-cut in an undirected graph using MA orderings presented above has a time complexity of $O(n(m + n \log n))$. The fastest known algorithm for finding a global min-cut in an undirected graph runs in $O(m \log^3 n)$ randomized time.

1.2 Random contraction

Random Contraction is another non-flow-based algorithm. It is based roughly on the premise that edges with large capacity are "unlikely" to be in a minimum cut. The idea is the following:

> **Random Contraction (Karger '93)**
>
> Until $|V| = 2$
> Pick edge (i, j) at random with probability proportional to its capacity.
> Contract (i, j).

Why could this work? Intuitively, the probability of contracting some edge in a minimum cut should be small, since the size of the cut is small. To make this precise, let S^* be some global min-cut. Let λ be the value of the global min-cut, that is $\lambda = u(\delta(S^*))$. Let $U_{tot} = \sum_{(i,j)\in E} u_{ij}$. Then the probability that no edge in $\delta(S^*)$ is contracted in the first iteration is $1 - \lambda/U_{tot}$.

Observe that $U_{tot} \geq \frac{n\lambda}{2}$. This follows because $u(\delta(i)) \geq \lambda$ for each vertex i, so that when we sum over all vertices and divide by 2 to eliminate double-counting of edges we get the right-hand side.

Using this bound we can write:

$$\Pr[\text{no edge in } \delta(S^*) \text{ contracted in first iteration}] = 1 - \frac{\lambda}{U_{tot}} \geq 1 - \frac{\lambda}{\frac{n\lambda}{2}} = 1 - \frac{2}{n}.$$

Let U^i_{tot} denote the sum of the capacities of remaining edges after i iterations of the algorithm. Let n_i denote the number of vertices after i iterations ($n_i = n - 1$). Observe that $U_{tot}^i \geq \frac{n_i \lambda}{2}$. This follows from the same reasoning as before. Thus, we can write a bound for the conditional probability that no edge in the global min-cut, $\delta(S^*)$, is contracted in the ith iteration given that no edge in $\delta(S^*)$ was contracted in the first $i-1$ iterations as follows:

$$1 - \frac{\lambda}{U_{tot}^{i-1}} \geq 1 - \frac{2}{n_{i-1}} = 1 - \frac{2}{n-i+1}.$$

Thus, the probability that no edge in $\delta(S^*)$ is contracted during the algorithm is greater than or equal to:

$$\prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) = \prod_{i=1}^{n-2} \left(\frac{n-i-1}{n-i+1}\right) = \prod_{j=n}^{3} \left(\frac{j-2}{j}\right) = \frac{1}{\binom{n}{2}} = \Omega(n^{-2}).$$

So, the probability that we contract no edge in the global min-cut during a single iteration of the randomized contraction algorithm is not huge, but it's also not insignificant. We'll see next time how we can make this into an algorithm that finds the global min-cut with high probability in a reasonable amount of time.

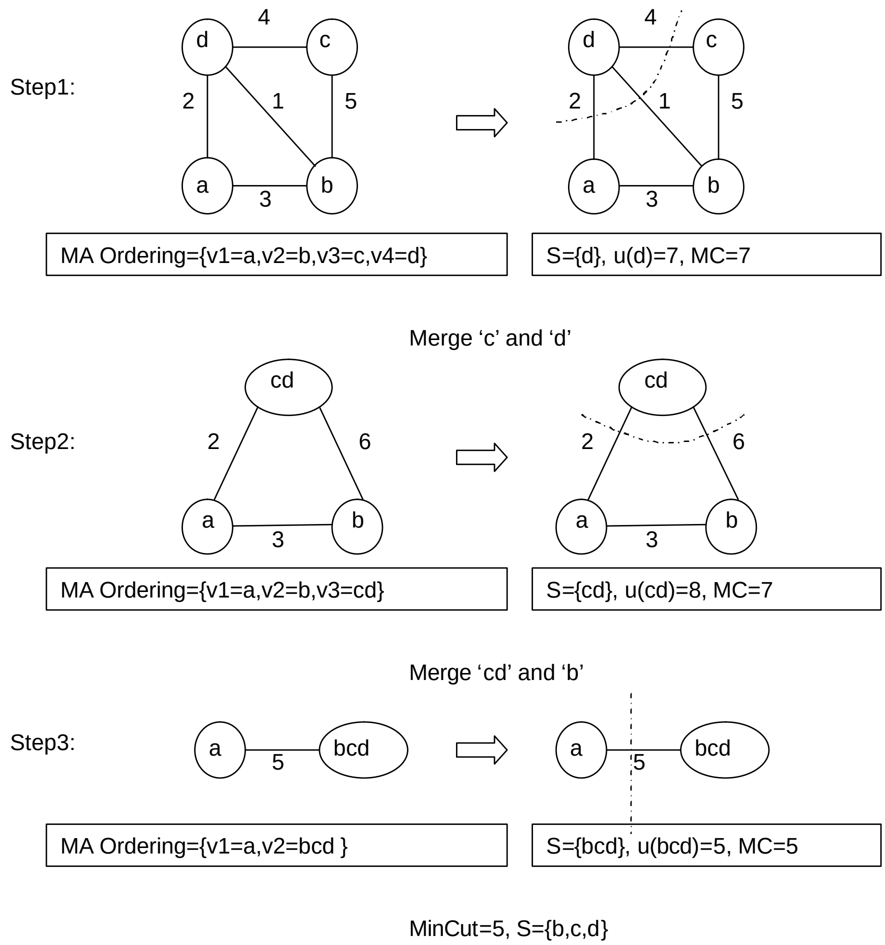

Figure 2: An example of the min-cut algorithm via MA orderings.