COM3105 Advanced Algorithms

Homework 2

Joffray Hargreaves

Question

(5 points)

Given a simple graph $G = (V, E)$ (no loops or multiple edges), consider the following deterministic algorithm:

- start with an arbitrary partition.
- If moving a vertex from one part to the other increases the number of crossing edges, we will move it.
- We do this until there is no such vertex left.

Why does this algorithm terminate?
What can you say about the number of edges crossing when the algorithm terminates?

Answer

Termination of the Algorithm:

- Let C be the set of edges crossing the partition.
- Let state i be the state of the partition after i moves have been made.
- For each state i, we iteratively check a vertex v_j (where $0 \leq j < |V|$) to see if moving it increases $|C|$.
- If moving a vertex v_j increases $|C|$ we make this move and move to state $i + 1$ and restart the same process.
- Checking each state i takes $\mathcal{O}(|V|)$ steps, where V is finite.
- As V is finite, for a simple graph, E is also finite, so the maximum $|C|$ is $|E|$.
- Therefore, in the worst case:
  - at state 0,
    - $|C| = 0$ (no edges in partition)
    - Adding $v_{|V|-1}$, increases $|C|$ by 1.
  - at state 1,
    - $|C| = 1$
    - Adding $v_{|V|-1}$, increases $|C|$ by 1.
  - at state i,
    - $|C| =$ i
    - Adding $v_{|V|-1}$, increases $|C|$ by 1.

- at state $|E|$,
  - $|C| = |E|$
  - No movement of vertex can increase $|C|$, as $|C| = |E|$.
  - Algorithm stops as no more edges to add.
- So $|E|$ edges being added to C and taking $|V|$ checks at each state.
- Hence the runtime is $\mathcal{O}(|E| * |V|)$.
- Both E and V are finite sets, so the algorithm terminates

Number of Crossing Edges at Termination.

Let the final partition of G be (A, B). For each vertex $v \in V$:

$$d_A(v) = |\{\, u \in A : (u,v) \in E \,\}|, \quad d_B(v) = |\{\, u \in B : (u,v) \in E \,\}|.$$

At termination, moving any vertex to the other part does not increase the number of crossing edges. Hence:

$$d_B(v) \geq d_A(v) \quad \text{for all } v \in A, \qquad \text{and} \qquad d_A(v) \geq d_B(v) \quad \text{for all } v \in B.$$

This means that for every vertex $v \in V$, at least half of its incident edges cross the partition:

$$d_C(v) \geq \tfrac{1}{2} d(v),$$

where,
$d(v) = d_A(v) + d_B(v)$ (i.e. the total degree of v),
$d_C(v) = \textit{number of degrees from } v \textit{ crossing the partition.}$

Summing over all vertices gives:

$$\sum_{v \in V} d_C(v) \geq \frac{1}{2} \sum_{v \in V} d(v)$$

Since each crossing edge contributes to $d_C(v)$ for exactly two vertices,

$$\sum_{v \in V} d_C(v) = 2|C|$$

Combining these two expressions:

$$2|C| \geq \frac{1}{2} \sum_{v \in V} d(v) = \frac{1}{2}(2|E|) = |E|$$

Therefore,

$$|C| \geq \frac{|E|}{2}$$

Conclusion. When the algorithm terminates, the resulting partition has at least half of all edges crossing between the two parts. Although this partition may be only a local maximum, it guarantees that at least $\frac{|E|}{2}$ edges are cut.