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 $\leq |V|$ steps to check, where V is finite.
- As V is finite, for a simple graph, E is aslo 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 $|E|$,
  - $|C| = |E|$
  - Adding $v_{|V|-1}$, cannot increase $|C|$ by 1, as $|C|$ is saturated.
  - 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, so the algorithm terminates

Number of Crossing Edges at Termination:
As the algorithm doesn't make any clever choices about which vertex to move, it is possible that the algorithm terminates in a local maximum, rather than a global one.

However, we can prove that the algorithm terminates with a minimum number of crossing edges.

Assume 2 sets A and B form the termination of partitioning of graph G.

For each vertex v in A, let $d_A(v)$ be the number of edges from v to other vertices in A, and let $d_B(v)$ be the number of edges from v to vertices in B.

$d_A(v) \leq d_B(v)$
The above must hold by definition, otherwise the algorithm has not terminated.
Similarly, for each vertex v in B, let $d_B(v)$ be the number of edges from v to other vertices in B, and let $d_A(v)$ be the number of edges from v to vertices in A.

$d_B(v) \leq d_A(v)$
we can summarise these by saying that, for every v in V:
$\textit{number of degrees from v crossing partion} \geq \frac{1}{2}d(v)$
where $d(v)$ is the total number of degrees of v

let $d_C(v) = \textit{number of degrees from v crossing partion}$
$d_C(v) \geq \frac{1}{2}d(v), \textit{ for all } v \in V$
Summing over all vertices in V:

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

Thus, when the algorithm terminates, the number of crossing edges is at least half the total number of edges in the graph.