CMPSCI 711: More Advanced Algorithms

Section 2-3: Matchings

Andrew McGregor

Last Compiled: April 29, 2012

Graph Matchings

Definition

A matching in graph $G = (V, E)$ is a subset of edges $M \subset E$ such that no two edges share an end point.

Problem

Find a matching M that maximizes $|M|$. If edges are weighted, we want to maximize $w(M) = \sum_{e \in M} w(e)$.

Unweighted Matching

- Let $M \leftarrow \emptyset$
- For each new edge e: add e to M if no edges in M share an endpoint with e

Theorem

Algorithm uses $O(n \log n)$ *space and returns a* 2 *approximation to the maximum weighted matching.*

Proof.

- Let $\text{OPT} = \{o_1, o_2, \ldots\}$ be set of edges in the optimal solution.
- Let M be final set of selected edges and note M is maximal.
- Since M is maximal, every edge in OPT shares an endpoint with at least one edge in M and hence $|\text{OPT}| \leq 2|M|$.

□

Weighted Matching Algorithm

- Let $M \leftarrow \emptyset$
- For each new edge e:
  - Let $C \subset M$ be set of edges sharing endpoint with e
  - If $w(e) \geq (1+\gamma)w(C)$ then $M \leftarrow \{M \setminus C\} \cup \{e\}$

Theorem

Algorithm uses $O(n \log n)$ space and returns a $(3 + 1/\gamma + 2\gamma)$ approx. to the max weighted matching. Setting $\gamma = 1/\sqrt{2}$ yields $(3 + 2\sqrt{2})$ approx.

Analysis Definitions

- An edge is *born* if it is ever included in M
- An edge is *bumped* if it was born but subsequently removed from M by a newer edge. The newer edge is *knocked out* the older edge.
- An edge is a *survivor* if it's born and not bumped.
- For a survivor $e \in S$, the *knock-out tree* $T(e)$ is

$$T(e) = C_1 \cup C_2 \cup C_3 \cup \dots$$

where C_1 is set of edges knocked out by e; C_2 is set of edges knocked out by edges in C_1; ...

Lemma

1. $w(T(S)) \leq w(S)/\gamma$
2. $w(\text{OPT}) \leq 2(1+\gamma)(w(T(S)) + w(S))$

where S is the set of survivor edges and $T(S) = \cup_{e \in S} T(e)$.

Exercise: Improve the analysis of second part of the lemma to show

$$w(\text{OPT}) \leq (1+\gamma)(w(T(S)) + 2w(S))$$

Proof of First Part of Lemma

- For each bumper e, $w(e)$ is at least $(1+\gamma)$ cost of bumped edges.
- Hence, for all i

$$w(C_i) \geq (1+\gamma)w(C_{i-1}) = \gamma w(C_{i-1}) + w(C_{i-1})$$

- Therefore,

$$\begin{aligned}
w(e) &\geq \gamma w(C_1) + w(C_1) \\
&\geq \gamma w(C_1) + \gamma w(C_2) + w(C_2) \\
&\geq \gamma w(C_1) + \gamma w(C_2) + \gamma w(C_3) + w(C_3) \\
&\geq \dots \\
&\geq \gamma w(T(e))
\end{aligned}$$

- Summing over survivors gives $w(S) \geq \gamma w(T(S))$

Proof of Second Part of Lemma

- Let $\text{OPT} = \{o_1, o_2, \ldots\}$ be set of edges in the optimal solution.
- We'll charge weights of OPT to edges in $A = \sum_{e \in S} T(e) \cup \{e\}$ such that no edge $e \in A$ gets charged much more $2(1+\gamma)w(e)$
- Say $e \in A$ is accountable to $o \in \text{OPT}$ if either $e = o$ or o was not born because e was in M when o arrived.
- If only one $e \in A$ is accountable to o: charge $w(o)$ to e
- If two $e_1, e_2 \in A$ are accountable to o: charge $\frac{w(e_i)w(o)}{w(e_1)+w(e_2)}$ to e_i. Note,

$$\frac{w(e_i)w(o)}{w(e_1)+w(e_2)} < \frac{w(e_i)(1+\gamma)(w(e_1)+w(e_2))}{w(e_1)+w(e_2)} = (1+\gamma)w(e_i)$$

- $e \in A$ is charged at most twice and each charge is $\leq (1+\gamma)w(e)$.

Open Research Questions

1. Can you beat the best known weighted matching approx. is 4.967.
2. Can you achieve any constant factor approximation when edges are inserted and deleted? Nothing is currently known.

Linear Sketches for Bipartiteness

- Can use linear sketches for connectivity to test bipartiteness.
- The *bipartite double cover* $D(G)$ of graph $G = (V, E)$ is formed by:
  - For each $v \in V$, $D(G)$ has two nodes v_1 and v_2
  - For each edge $(u, v) \in E$, $D(G)$ has two edges (u_1, v_2) and (u_2, v_1)
- *Algorithm:* Apply connectivity sketch to G and $D(G)$. If number of connected components in $D(G)$ is twice the number of connected components in G then return `bipartite`.
- *Analysis:*
  - For simplicity assume G is connected (otherwise we can apply analysis on individual connected components).
  - *Lemma:* $D(G)$ has two connected components iff G is bipartite.

Proof of Lemma

- Suppose G not bipartite and therefore there exists an odd cycle.
  - Let u be a node in the odd cycle.
  - Since cycle is odd, there's a path from u_1 to u_2 in $D(G)$.
  - Since G connected, all nodes in $D(G)$ are reachable from u_1 or u_2.
  - Hence there is a path between any two nodes in $D(G)$.
- Suppose $D(G)$ has only one connected component.
  - There exists a path from u_1 to u_2 in $D(G)$
  - This corresponds to an odd length cycle in G so G can't be bipartite.