1 Another Easy 2-Approximation

Let $G = (V, E)$ be an undirected graph.

- A cut $(L, R)$ of $G$ is a partition of $V$ into two sets $L$ and $R = V - L$.
- The size of a cut is the number of edges that go between $L$ and $R$.
- There are two natural problems about cuts.
- $MIN$-$CUT$ problem: find a cut of minimum size in $G$.
- $MAX$-$CUT$ problem: find a cut of maximum size in $G$.
- One of these is easy! If we fix vertices $s$ and $t$ in $G$, size of a minimum cut that separates $s$ and $t$ equals size of a maximum flow between them.
- Max flow can be solved by linear programming or other methods in polynomial time, yielding a polytime solution to minimum cut.
- What about maximum cut? Can show that it is an NP optimization problem!
- Standard reduction comes from “not all equal” 3-SAT (is there a satisfying assignment in which every clause has at least 1 true and 1 false literal?)
- Let’s look at approximability.

2 Proposed Algorithm Number 1

This is a “local search” approach. Call it LOCAL-CUT.

1. Start with an arbitrary cut $(L, R)$ of $G$.
2. While there a vertex $v$ such that moving $v$ to the other side of the cut increases its size, move $v$ to the other side.
3. Finally, report the final cut.

Is this algorithm any good?

- Correctness: at every point, algo maintains a valid cut.
• **Cost:** every move increases the cut size.

• \(|E|\) upper-bounds max possible cut size.

• Hence, algo terminates after at most \(|E|\) steps.

• Each step naively takes time \(O(|V| + |E|)\), so total cost is \(O(|E|(|V| + |E|))\).

• What about approximation bound?

**Lemma:** LOCAL-CUT is a 2-approximation.

• As we said before, upper bound on size of max cut is trivially \(|E|\).

• Will show that LOCAL-CUT finds cut of size at least \(|E|/2\).

• Consider final cut \((L, R)\) produced by LOCAL-CUT.

• Let \(v\) be any vertex in \(V\).

• We claim that at least half the edges incident on \(v\) cross the cut \((L, R)\).

• Suppose not; then moving \(v\) to other side of the cut would increase number of crossing edges, so LOCAL-CUT would have moved it.

• Let \(d(v)\) be degree of \(v\) (# edges incident on \(v\)).

• By above claim, size of cut \((L, R)\) is at least

\[
\frac{1}{2} \sum_{v \in V} \frac{d(v)}{2}.
\]

• (extra \(1/2\) because we count each crossing edge once for each of its endpts)

• But every edge in \(E\) touches exactly two vertices, so we have

\[
|E| = \frac{1}{2} \sum_{v \in V} d(v).
\]

• Conclude that size of \((L, R)\) is at least \(|E|/2\). QED

3 Proposed Algorithm Number 2

Here's another easy approach, called GREEDY-CUT.

• Start with empty sets \(L, R\).

• Pick any vertex \(v \in V\) and put it in \(L\).

• For remaining vertices \(u \in V\) in any order:

  • Let \(s_L\) be size of cut \((L \cup \{u\}, R)\).
  • Let \(s_R\) be size of cut \((L, R \cup \{u\})\).
If \( s_L > s_R \), add \( u \) to \( L \); otherwise, add it to \( R \).

- Again, clearly ends up with correct cut, and runs in time polynomial in \( |G| \).

**Lemma:** GREEDY-CUT is a 2-approximation.

- Again, \( |E| \) is an upper bound on size of maximum cut.
- Will prove that GREEDY-CUT produces a cut of size at least \( |E|/2 \).
- For each edge \( e \in E \), say that second endpoint of \( e \) that occurs in chosen vertex order is “responsible” for \( e \).
- For each vertex \( v \in V \), let \( r(v) \subseteq E \) be set of edges for which \( v \) is responsible.
- When \( v \) is placed, each edge in \( r(v) \) adds either 1 or 0 to cut size (1 if it crosses, 0 otherwise)
- Best placement for \( v \) causes at least half of edges in \( r(v) \) to cross cut, hence adds at least \( |r(v)|/2 \) to cut size.
- Conclude that size of cut is at least

\[
\sum_{v \in V} |r(v)|/2.
\]

- But every edge has exactly one responsible vertex, so

\[
\sum_{v \in V} |r(v)| = |E|.
\]

- Conclude that final cut has size at least \( |E|/2 \). QED

4 A Few Final Words

- Best known approximation ratio for MAX-CUT (AFAIK) is 1.138217, from SDP algorithm of Goemans and Williamson (1995).
- Better approx results for various restrictions of the problem
- Has been proven inapproximable within 1.0625 unless P = NP (Hastad, 2001).
- If Unique Games Conjecture holds, G-W bound is tight (Khot, Kindler, Mossel, and O’Donnell 2005).