Local Search

The Maximum Cut Problem

Algorithms: Design and Analysis, Part II

The Maximum Cut Problem

Input: an undirected graph $G=(V,E)$.

Goal: a cut (A,B) – a partition of V into two non-empty sets – that maximizes the number of crossing edges.

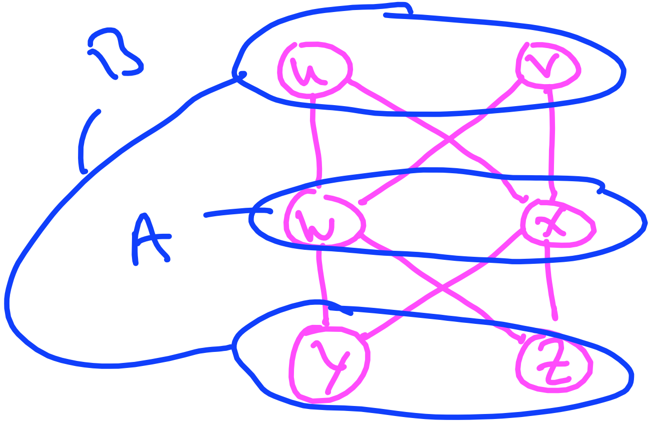
Sad fact: NP-complete.

Computationally tractable special case: bipartite graphs (i.e., where there is a cut such that all edges are crossing)

exercise: solve in linear time via breadth-first search

Quiz

Question: What is the value of a maximum cut in the following graph?



(A) 4

(B) 6

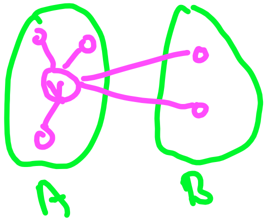
(C) 8

(D) 10

A Local Search Algorithm

Notation: for a cut (A,B) and a vertex v, define

$c_v(A,B)$ = # of edges incident on v that cross (A,B)
$d_v(A,B)$ = # of edges incident on v than don't cross (A,B)



Local search algorithm

(1) let (A,B) be an arbitrary cut of G.
(2) while there is a vertex v with $d_v(A,B) > c_v(A,B)$:
- move v to other side of the cut [key point: increases number of crossing edges by $d_v(A,B) - c_v(A,B) > 0$]

(3) return final cut (A,B)

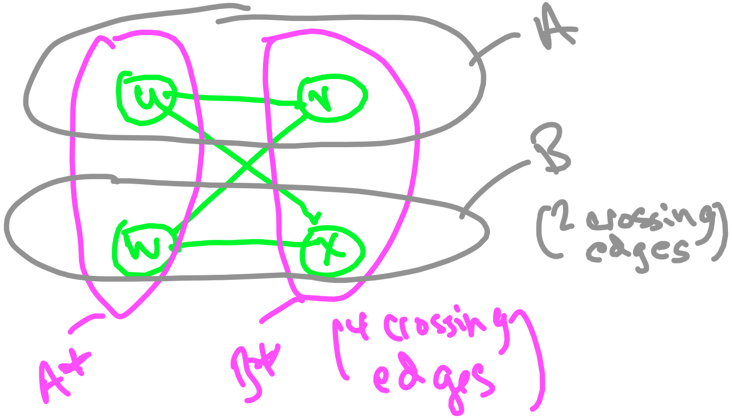
Note: terminates within $\binom{n}{2}$ iterations [+ hence in polynomial time].

Tim Roughgarden

Performance Guarantees

Theorem: this local search algorithm always outputs a cut in which the number of crossing edges is at least 50% of the maximum possible. (even $\geq 50\%$ of $|E|$)

Tight example:



Cautionary Point: expected number of crossing edges of a random cut is already $\frac{1}{2}|E|$.

Proof: Consider a random cut (A,B).

For edge $e \in E$, define $X_e = \begin{cases} 1 & \text{if } e \text{ crosses } (A,B) \\ 0 & \text{otherwise} \end{cases}$

We have $E[X_e] = \Pr[X_e = 1] = 1/2$.

So $E[\#\text{ crossing edges}] = E[\sum_e X_e] = \sum_e E[X_e] = \frac{|E|}{2}$.

LIN EXP

qed.

Tim Roughgarden

Proof of Performance Guarantee

Let (A,B) be a locally optimal cut.
Then, for every vertex v, $d_v(A,B) \le c_v(A,B)$.

counts each crossing edge twice

Summing over all $v \in V$: $\sum_{v \in V} d_v(A,B) \le \sum_{v \in V} c_v(A,B)$

counts each non-crossing edge twice

So: $2 \cdot$ [# of non-crossing edges] $\le 2 \cdot$ [# of crossing edges]

$\Rightarrow 2 \cdot |E| \le 4 \cdot$ [# of crossing edges]

$\Rightarrow$ # of crossing edges $\ge \frac{1}{2}|E|$.

QED!

Tim Roughgarden

The Weighted Maximum Cut Problem

Generalization: each edge $e \in E$ has a nonnegative weight w_e, want to maximize total weight of crossing edges.

Notes:

(1) local search still well defined

(2) performance guarantee of 50% still holds for locally optimal cuts [you check!] (also for a random cut)

(3) no longer guaranteed to converge in polynomial time [non-trivial exercise]