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.

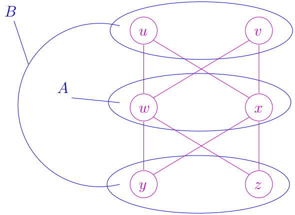
Sad fact: NP-complete.

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

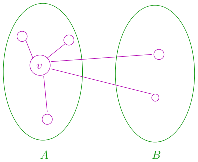
B) 6

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 that 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)

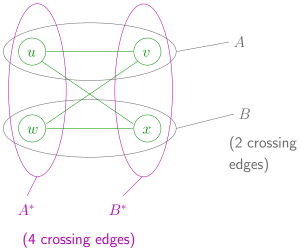
Note: Terminates within $\binom{n}{2}$ iterations [+ hence in polynomial time].

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 50% of $|E|$)

Tight example:



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

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 $\mathrm{E}[X_e]=\Pr[X_e = 1]=1/2$.
So $\mathrm{E}[\#\text{ crossing edges}]=\mathrm{E}[\sum_e X_e]=\sum_e \mathrm{E}[X_e]=|E|/2$. QED

Proof of Performance Guarantee

Let (A, B) be a locally optimal cut. Then, for every vertex v, $d_v(A, B) \leq c_v(A, B)$. Summing over all $v \in V$:

$$\boxed{\sum_{v \in V} d_v(A, B)} \leq \boxed{\sum_{v \in V} c_v(A, B)}$$

counts each non-crossing edge twice

counts each crossing edge twice

So:

2·[# of non-crossing edges] $\leq$ 2·[# of crossing edges]

2·$|E| \leq$ 4·[# of crossing edges]

# of crossing edges $\geq \frac{1}{2}|E|$ QED!

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]