4.1 The Probabilistic Method

4.1.1 The Max Cut Problem

We now consider another example that demonstrates the use of the probabilistic method: the maxcut problem.

**Input:** A graph $G = (V, E)$.

**Output:** Find a subset $S \subseteq V$ such that the number of edges between $S$ and $\overline{S} = V \setminus S$ is maximized.

Determining the maxcut of a graph is an NP-hard problem. However, a simple application of the probabilistic method gives us a lower bound on the size of the maximum cut.

**Claim 4.1** In any graph, there exists a cut with at least half the edges crossing it.

**Proof:** Let $S \subseteq V$ be a random subset that includes every vertex independently with probability $\frac{1}{2}$.

Let $X$ be the size of the cut, which is the number of edges crossing the cut.

$$X = |E(S, \overline{S})|$$

For each edge $e$, define an indicator variable $X_e$ as

$$X_e = \begin{cases} 1 & \text{if } e \in E(S, \overline{S}) \\ 0 & \text{otherwise} \end{cases}$$

Since $X$ is the number of crossing edges: $X = \sum_{e \in E} X_e$.

Moreover, we have:

$$\mathbb{E}[X_e] = \Pr[e \in E(S, \overline{S})] \cdot 1 + \Pr[e \notin E(S, \overline{S})] \cdot 0$$

Therefore,

$$\mathbb{E}[X_e] = \Pr[e \text{ is cut}] = \Pr[\text{one endpoint of } e \text{ lies in } S \text{ and the other does not}] = \frac{1}{2}$$

By linearity of expectation,

$$\mathbb{E}[X] = \sum_{e \in E} \mathbb{E}[X_e] = \frac{|E|}{2}$$
In the sample space, there must exist a point at which $X$ takes value at least $\mathbb{E}[X]$. Thus, there exists a cut with at least half the edges crossing it.

### 4.1.1.1 Constructing good solutions - Markov’s inequality

The previous section proved that there exists a cut with at least half the edges crossing it without telling how to construct such a solution. In fact, the probability of hitting one may be vanishingly small. In this section, we show that the randomized algorithm which assigns a vertex randomly to a cut produces good solutions - having more than $\frac{|E|}{4}$ edges - with high probability. To bound the chance of getting good solutions, we use Markov’s Inequality (see last lecture for the theorem).

Let $Y = |E| - X$, the number of uncut edges. Then $Y$ is nonnegative and

$$
\mathbb{E}[Y] = |E| - \mathbb{E}[X] = \frac{|E|}{2}
$$

By Markov’s Inequality,

$$
\Pr[Y \geq \alpha \mathbb{E}[Y]] = \Pr[Y \geq \frac{\alpha |E|}{2}] \leq \frac{1}{\alpha}
$$

Choose $\alpha = \frac{3}{2}$, we have

$$
\Pr[Y \geq \frac{3}{4} |E|] \leq \frac{2}{3}
$$

or in other words,

$$
\Pr[X > \frac{1}{4} |E|] \geq \frac{1}{3}
$$

Therefore, a random cut will have more than $\frac{1}{4}$ of the edges with the probability at least $\frac{1}{3}$.

### 4.1.1.2 Method of conditional probabilities

There is a simple way to derandomize the algorithm for maxcut using a greedy algorithm. Assign an order on the vertices: $v_1, \ldots, v_n$. Consider one vertex at a time. We have

$$
\frac{1}{2} \cdot |E| = \mathbb{E}[X] = \Pr[v_1 \in S] \cdot \mathbb{E}[X|v_1 \in S] + \Pr[v_1 \notin S] \cdot \mathbb{E}[X|v_1 \notin S].
$$

Thus, either $\mathbb{E}[X|v_1 \in S]$ or $\mathbb{E}[X|v_1 \notin S] \geq \frac{|E|}{2}$. Include $v_1$ in $S$ only if $\mathbb{E}[X|v_1 \in S] \geq \frac{|E|}{2}$. Do the same for all vertices one by one, moving down the tree of all possible vertex settings. After the last vertex, we get to a leaf where the expected value of the cut is at least $\frac{|E|}{2}$. This way, we have a deterministic algorithm to find a cut of size at least $\frac{|E|}{4}$. But we still need to calculate the conditional expectation $\mathbb{E}[X|S \cap \{v_1, \ldots, v_k\}]$ efficiently. This is easy to do and is left as an exercise.

### 4.2 Crossing number of a graph

Let $G = (V, E)$ be a graph. The crossing number of $G$, denoted $cr(G)$ is the minimum number of edge crossings when $G$ is drawn optimally in the plane. By definition, $G$ is planar iff $cr(G) = 0$.

Let $m = |E|$ and $n = |V|$. By Euler’s formula, we get the following lemma:
Lemma 4.2 if $G$ is planar, then $m \leq 3n - 6$. Hence $cr(G) > 0$ if $m > 3n - 6$.

The next claim uses the lemma to derive a lower bound on the crossing number of a graph.

Claim 4.3 For any graph $G$, $cr(G) \geq m - 3n + 6$.

Proof: Draw $G$ optimally in the plane and remove edges contributing to crossings one by one. As each edge is removed, $cr(G)$ decreases by at least 1. When the number of edges remaining reaches $3n - 6$, we know by the lemma above that the remaining graph has no more crossings, i.e $cr(G) = 0$. Hence, $cr(G) - 0 \geq m - (3n - 6)$. We get $cr(G) \geq m - 3n + 6$.

For the complete graph $K_n$, this bound is loose. The crossing number of $K_n$ is conjectured to be $\Theta(n^4)$ while the above bound gives $\Omega(n^2)$ ($|E| = \binom{n}{2} = \Theta(n^2)$). The next claim gives an $\Omega(n^4)$ lower bound for $cr(K_n)$.

Claim 4.4 If $m \geq 4n$, then $cr(G) \geq \frac{m^3}{64n^2}$.

Proof: Fix an optimal drawing of $G$. Let $G_p$ be the induced subgraph formed by including every vertex of $G$ with probability $p$. Let $n_p$, $m_p$ and $c_p$ be the number of vertices, number of edges, and number of crossings in $G_p$, respectively. Hence,

- $E[n_p] = pm$.
- $E[m_p] = p^2 m$.
- $E[c_p] = p^4 cr(G)$.

From the previous claim, we have $c_p \geq m_p - 3n_p + 6$. By linearity of expectation, $E[c_p] \geq E[m_p] - 3E[n_p]$. Replacing the expectations by their values, we get

$$p^4 cr(G) \geq mp^2 - np \Rightarrow cr(G) \geq \frac{m}{p^2} - \frac{n}{p^3}$$

Setting $p$ to $\frac{4n}{m}$, we get $cr(G) \geq \frac{m^3}{16n^2} - \frac{3m^3}{64n^2} = \frac{m^3}{64n^2}$. Note that the condition $m \geq 4n$ is needed to make sure that $p \leq 1$.

### 4.3 Monotone Circuits For The Majority Function

A boolean circuit computes a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$. It has $n$ inputs $\{x_1, x_2, \ldots, x_n\}$ and one output. There are also arbitrary 2-input gates. Each gate can compute any of the two-input functions. The depth of the circuit is the maximum lengths of directed path from any input to the output. The size of the circuit is the number of gates.

A function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is a monotone function if whenever $f(x_1, x_2, \cdots, x_n) = 1$ and $\forall i, y_i \geq x_i$, $f(y_1, y_2, \cdots, y_n) = 1$.

A monotone circuit is one with all gates computing monotone functions.

We are interested in designing small size/small depth circuits for the majority function on $n$ bits.

$$Maj(x_1, \cdots, x_n) = \begin{cases} 1 & \text{if } |\{i : x_i = 1\}| > \frac{n}{2} \\ 0 & \text{otherwise} \end{cases}$$
It is easy to design a boolean circuit for majority that has size $O(n)$ and depth $O(\log n)$. However, we are only interested in the monotone circuits, which can compute only monotone functions. More interesting analysis for this problem will be given next lecture.