Topics Covered

1. Examples of Approximation Algorithms
2. Generalization
3. Hardness of Approximation

1 Examples of Approximation Algorithms

**Approximate Solution for VertexCover**  Given an undirected graph $G$, the MIN-VERTEXCOVER problem is to find a smallest vertex cover for $G$. The following is an approximation algorithm for MIN-VERTEXCOVER (Theorem 10.1 in Sipser):

On input $(G)$:

1. While not all edges are marked:
   
   (a) Find an edge $e$ not touched by any marked edge, and mark $e$.

2. Output the set of all vertices that are endpoints of marked edges.

First, note that this algorithm takes polynomial time with respect to $|G|$. It correctly outputs a vertex cover because every edge is either marked or touched by a marked edge, and the vertex cover includes both endpoints of every marked edge. Moreover, the output vertex cover is at most twice as large as a smallest vertex cover. This is because the number of vertices in the cover is twice the number of marked edges, and the number of marked edges is at most the size of a minimal vertex cover. In particular, no marked edges can touch one another, so any vertex cover needs to include at least one endpoint of every marked edge.

**Approximate Solution for 3SAT**  Given a 3CNF such that every clause is the disjunction of three distinct literals, the MAX-3SAT problem is to find an assignment satisfying as many clauses as possible. Here is a randomized approximation for MAX-3SAT:
On input \( \phi \):

1. For each variable \( x_i \), give it a random assignment of ‘true’ or ‘false’.

This algorithm takes probabilistic polynomial time. To analyze correctness, we consider the expected number of clauses that are satisfied. For each clause \( \phi_i \) out of a total of \( m \) clauses, let the indicator random variable \( v_i \) be 1 if \( \phi_i \) is satisfied and 0 otherwise. Then \( \Pr[v_i = 1] = 1 - \Pr[v_i = 0] = 1 - \frac{1}{8} = \frac{7}{8} \). Overall, the expectation \( E[\# \text{ of satisfied clauses}] = E[\sum_{i=1}^{m} v_i] = \sum_{i=1}^{m} E[v_i] = \sum_{i=1}^{m} \frac{7}{8} = \frac{7}{8}m \). At most, \( m \) clauses of the 3CNF are satisfied, so this algorithm produces an assignment satisfying at least \( \frac{7}{8} \) of the maximum number of simultaneously satisfiable clauses.

2 Generalization

In general, there are three variations of problems: decision, search, and optimization. Let \( R \) be an indicator relation and let \( f \) be a “quality” function.

- **Decision problems**: On input \( (x, v) \), accept if there exists a \( y \) such that \( R(x, y) \) and \( f(y) = v \). As an example, consider the VERTEXCOVER problem. Then \( R((G), (C)) = 1 \) if \( C \) is a vertex cover of \( G \), and \( f((C)) = |C| \).

- **Search problems**: On input \( (x, v) \), find a \( y \) such that \( R(x, y) \) and \( f(y) = v \).

- **Optimization problems**: On input \( x \), find a \( y \) such that \( R(x, y) \) and \( f(y) \) is optimized. Note that \( f(y) \) can either be minimized (e.g. MIN-VERTEXCOVER) or maximized (e.g. MAX-3SAT).

We use approximation algorithms to model optimization problems. An \( \alpha \)-approximation problem on input \( x \) is to find a \( y \) such that:

- \( f(y) \leq \alpha \cdot \text{OPT} \) if it’s a minimization problem,

- \( f(y) \geq \frac{1}{\alpha} \cdot \text{OPT} \) if it’s a maximization problem.

For example, we’ve described a 2-approximation for MIN-VERTEXCOVER and an \( \frac{8}{7} \)-approximation for MAX-3SAT.
3 Hardness of Approximation

PCP (Probabilistically Checkable Proof) Theorem  For all constants $\varepsilon > 0$, for all $L \in \text{NP}$, there exists a polynomial-time reduction $f$ such that for all $x$:

- $x \in L$ implies $f(x) \in \text{3SAT}$
- $x \notin L$ implies $f(x)$ is a 3CNF where the fraction of clauses that can be simultaneously satisfied is smaller than $(\frac{7}{8} + \varepsilon)$.

**Corollary**  If a $1/(\frac{7}{8} + \varepsilon)$-approximation exists for 3SAT for any $\varepsilon$, then P = NP.

**Proof of Corollary**  Suppose such an approximation for 3SAT exists. Let $L$ be a language in NP. Then $L$ can be decided in polynomial-time by the following algorithm:

1. Compute a 3CNF $\phi = f(x)$ via the 3SAT reduction.
2. Run the approximation algorithm on $\phi$.
3. If the output assignment $a_1, \ldots, a_n$ satisfies at least a $(\frac{7}{8} + \varepsilon)$ fraction of the clauses, accept.
4. Otherwise, reject.

If $x \in L$, then it is possible to satisfy all clauses, and the approximation algorithm finds an assignment satisfying at least a $(\frac{7}{8} + \varepsilon)$ fraction of the clauses. On the other hand, if $x \notin L$, then no assignment can satisfy all of the clauses. The approximation can only produce assignments satisfying a smaller fraction than $(\frac{7}{8} + \varepsilon)$ of the clauses.

MAX-CLIQUE  The MAX-CLIQUE problem is given input $(G)$, to output $(C)$ such that $C$ is a clique in $G$, as large as possible. We claim that if P \neq NP, then for all $\varepsilon > 0$, there is no $1/(\frac{7}{8} + \varepsilon)$-approximation for MAX-CLIQUE.

Recall the reduction $f$ from 3SAT to CLIQUE. On input $\phi$ with $m$ clauses, we make three vertices for every clause and connect vertices that don’t contradict each other and are in different clauses. If $\phi$ is satisfiable, then $G$ has a clique of size $m$. If fewer than $(\frac{7}{8} + \varepsilon)m$ of the clauses can be simultaneously satisfied, then the largest clique in $G$ is smaller than a $(\frac{7}{8} + \varepsilon)m$. This is because a clique can have one vertex from each clause, and no vertices can contradict one another. That is, every clause with a vertex in the clique is satisfied by the
assignment. As an approximation for MAX-CLIQUE yields an approximation for 3SAT, it follows from the previous corollary that if \( P \neq NP \), there can be no \( 1/(\frac{7}{8} + \varepsilon) \)-approximation for MAX-CLIQUE.

**MIN-VertexCover** Consider a similar reduction, but to \textsc{IndependentSet} instead of \textsc{CLIQUE}. Recall the \textsc{IndependentSet} reduction, where given a CNF \( \phi \) with \( m \) clauses, \( G = (V, E) \) has \( |V| = 3m \). If \( \phi \) is satisfiable, \( G \) has an independent set of size \( m \), or equivalently, a vertex cover of size \( 2m = \frac{32}{10}m \). If \( \phi \) is not satisfiable, then a \( 1/(\frac{7}{8} + \varepsilon) \)-approximation would find an assignment with fewer than \( (\frac{7}{8} + \varepsilon)m \) satisfied clauses, which corresponds to an independent set with fewer than \( (\frac{7}{8} + \varepsilon)m \) vertices. A vertex cover would then be of size at least \( 3m - \frac{7}{8}m - \varepsilon m \). If we let \( \varepsilon = \frac{1}{16} \), for example, a minimal vertex cover would have size at least \( \frac{33}{32}m \).

Thus, if we have a \( \frac{33}{32} \)-approximation for \textsc{MIN-VertexCover}, we can decide any \( L \in NP \) in polynomial time:

On input \( x \):

1. Compute \( \phi = f(x) \) using \( f \) from the PCP Theorem.
2. Compute \( G \) from \( \phi \) via the reduction from 3SAT to \textsc{IndependentSet}.
3. Find a \( \frac{33}{32} \)-approximation vertex cover in \( G \).
4. If the size of the cover is smaller than \( \frac{33}{32} \), accept.
5. Otherwise, reject.