Today’s Lecture:

- Approximation algorithm for set cover
- AP-reducibility
  - L-reduction technique
Minimum Set-Cover:

- **INSTANCE**: Collection $C$ of subsets of a finite set $S$.

- **SOLUTION**: A set cover for $S$, i.e., a subset $C'$ of $C$ such that every element in $S$ belongs to at least one member of $C'$

- **MEASURE**: $|C'|$

Johnson’s algorithm:

- Polynomial-time logarithmic approximation algorithm for MINIMUM SET COVER

begin

$U:=S$; $C':=\emptyset$;
for any $c_i$ do $c'_i := c_i$;
repeat

$i:=\text{index of } c' \text{ with maximum cardinality};$
insert $c_i$ in $C'$;
$U := U \setminus \{\text{elements of } c'_i \}$;
delete all elements of $c_i$ from all $c'$;

until $U:=\emptyset$
end.
Performance Guarantee:

- We prove that the performance ratio of the algorithm is at most $\frac{1}{i} \leq k \leq i$ where $k$ is the maximum cardinality of the sets in $C$.
- Let $a_1, \ldots, a_{|C'|}$ be the sequence of indices obtained by the algorithm.
- Let $c_i^j$ be the surviving part of $c_i$ before index $a_j$ has been chosen.
- The intersection of $c_i$ and $c_{a_j}^j$ is equal to $c_i^j - c_{i+1}^j$.
- $l_i$ denote largest index $j$ such that $c_i^j$ is not empty.

Theorem 1:

For any $i$, $H(|c_i|) \geq \sum_{1 \leq j \leq |C'|} (|c_i \cap c_{a_j}^j| / |c_{a_j}^j|)$ where $H(n) = \sum_{1 \leq i \leq n} (1/i)$.

Proof:

$$\sum_{i \in S \subseteq |C'|} (|c_i \cap c_{a_j}^j| / |c_{a_j}^j|) = \sum_{i \in S \subseteq |C'|} (|c_i^j - |c_i^{j+1}| / |c_{a_j}^j|)$$

$$\leq \sum_{i \in S \subseteq |C'|} (|c_i^j| - |c_i^{j+1}| / |c_{a_j}^j|)$$

$$= \sum_{i \in S \subseteq |C'|} \sum_{1 \leq i \leq k} (1 / |c_i^j|)$$

$$\leq \sum_{i \in S \subseteq |C'|} \sum_{1 \leq i \leq k} (1 / |c_i^{j+1}|)$$

$$\leq \sum_{i \in S \subseteq |C'|} (H(|c_i^j|) - H(|c_i^{j+1}|))$$

$$= H(|c_i|) = H(|c_i|)$$
Theorem 2:

For any set cover $C''$, 
\[ \sum_{c_i \in C''} \sum_{1 \leq j \leq |C'|} \left( \frac{|c_i \cap c^j |}{|c^j |} \right) \geq |C'| \]

Proof:

\[
\sum_{c_i \in C''} \sum_{1 \leq j \leq |C'|} (|c_i \cap c^j | / |c^j |) = \sum_{1 \leq j \leq |C'|} (1/|c^j |) \sum_{c_i \in C''} (|c_i \cap c^j |) \\
\geq \sum_{1 \leq j \leq |C'|} (|c^j | / |c^j |) \\
= |C'| 
\]

Finally:

We have $H(|c_i |) \geq \sum_{1 \leq j \leq |C'|} (|c_i \cap c^j | / |c^j |)$

and $\sum_{c_i \in C''} \sum_{1 \leq j \leq |C'|} (|c_i \cap c^j | / |c^j |) \geq |C'|$

That will imply $\sum_{c_i \in C''} H(|c_i |) \geq |C'|$.

Since $|c_i | \leq k$, we have that $\sum_{c_i \in C''} H(K) \geq |C'|$

that is $|C''| \times H(K) \geq |C'|$.

Since $H(K) \leq \ln k + 1 \leq \ln n + 1$, we have the desired performance ratio.
Reducibility and NPO problems

\[ x, r \xrightarrow{f(x), r'} \]

\[ g(x, y) \xrightarrow{y} \]

\( r \)-approximate solution of \( x \) \hspace{2cm} \( r' \)-approximate solution of \( f(x) \)

AP-reducibility

- \( P_1 \) is AP-reducible to \( P_2 \) if two functions \( f \) and \( g \) and a constant \( c \geq 1 \) exist such that:
  - For any instance \( x \) of \( P_1 \) and for any \( r \), \( f(x, r) \) is an instance of \( P_2 \)
  - For any instance \( x \) of \( P_1 \), for any \( r \), and for any solution \( y \) of \( f(x, r) \), \( g(x, y, r) \) is a solution of \( x \)
  - For any fixed \( r \), \( f \) and \( g \) are computable in polynomial time
  - For any instance \( x \) of \( P_1 \), for any \( r \), and for any solution \( y \) of \( f(x, r) \), if \( R(f(x, r), y) \leq r \), then \( R(x, g(x, y, r)) \leq 1 + c(r-1) \)
Basic properties

- **Theorem:** If $P_1$ is AP-reducible to $P_2$ and $P_2$ is in APX, then $P_1$ is in APX
  - If $A$ is an $r$-approximation algorithm for $P_2$ then
    $g(x, A(f(x), r))$ is a $(1 + c(r-1))$-approximation algorithm for $P_1$

- **Theorem:** If $P_1$ is AP-reducible to $P_2$ and $P_2$ is in PTAS, then $P_1$ is in PTAS
  - If $A$ is a polynomial-time approximation scheme for $P_2$ then
    $g(x, A(f(x, r'), r'))$ is a polynomial-time approximation scheme for $P_1$, where $r' = 1 + (r-1)/c$

L-reducibility

- $P_1$ is L-reducible to $P_2$ if two functions $f$ and $g$ and two constants $a$ and $b$ exist such that:
  - For any instance $x$ of $P_1$, $f(x)$ is an instance of $P_2$
  - For any instance $x$ of $P_1$, and for any solution $y$ of $f(x)$, $g(x, y)$ is a solution of $x$
  - $f$ and $g$ are computable in polynomial time
  - For any instance $x$ of $P_1$, $m^*(f(x)) \leq am^*(x)$
  - For any instance $x$ of $P_1$ and for any solution $y$ of $f(x)$, $|m^*(x) - m(x, g(x, y))| \leq b|m^*(f(x)) - m(f(x), y)|$
Basic property of L-reductions

**Theorem:** If $P_1$ is L-reducible to $P_2$ and $P_2$ is in PTAS, then $P_1$ is in PTAS
- Relative error in $P_1$ is bounded by $ab$ times the relative error in $P_2$

However, in general, it is not true that if $P_1$ is L-reducible to $P_2$ and $P_2$ is in APX, then $P_1$ is in APX
- The problem is that the relation between $r$ and $r'$ may be non-invertible

Inapproximability of clique

**Theorem:** MAXIMUM 3-SAT is L-reducible to MAXIMUM CLIQUE
- $f(C,U)$ is the graph $G(V,E)$ where $V=\{(l,c) : l \text{ is in clause } c\}$ and $E=\{((l,c),(l',c')) : l \neq l' \text{ and } c \neq c'\}$
- $g(C,U,V')$ is a truth-assignment $t$ such that $t(u)$ is true if and only if a clause $c$ exists for which $(u,c)$ is in $V'$
- $a=b=1$
  - $t$ satisfies at least $|V'|$ clauses
  - optimum measures are equal

**Corollary:** MAXIMUM CLIQUE does not belong to APX
Inapproximability of independent set

- **Theorem:** MAXIMUM CLIQUE is AP-reducible to MAXIMUM INDEPENDENT SET
  - \( f(G=(V,E)) = G^c=(V,V^2-E) \), which is called the complement graph
  - \( g(G,U)=U \)
  - \( c=1 \)
    - Each clique in \( G \) is an independent set in \( G^c \)

- **Corollary:** MAXIMUM INDEPENDENT SET does not belong to APX

Inapproximability of 2-satisfiability

- **Theorem:** MAXIMUM 3-SAT is L-reducible to MAXIMUM 2-SAT
  - \( f \) transforms each clause \( x \text{ or } y \text{ or } z \) into the following set of 10 clauses where \( i \) is a new variable:
    - \( x, y, z, i, \text{ not } x \text{ or } \text{ not } y, \text{ not } x \text{ or } \text{ not } z, \text{ not } y \text{ or } \text{ not } z, \text{ not } x \text{ or } \text{ not } i, \text{ not } y \text{ or } \text{ not } i, \text{ not } z \text{ or } \text{ not } i \)
  - \( g(C,t)=\text{restriction of } t \text{ to original variables} \)
  - \( a=13, b=1 \)
    - \( m^*(f(x))=6|C|+m^*(x) \leq 12m^*(x)+m^*(x)=13m^*(x) \)
    - \( m^*(f(x))-m(f(x),t) \leq m^*(x)-m(x,g(C,t)) \)

- **Corollary:** MAXIMUM 2-SAT is not in PTAS
MAXIMUM NOT-ALL-EQUAL SAT

- INSTANCE: CNF Boolean formula, that is, set $C$ of clauses over set of variables $V$

- SOLUTION: A truth-assignment $f$ to $V$

- MEASURE: Number of clauses that contain both a false and a true literal

Inapproximability of NAE 2-satisfiability

- **Theorem:** MAXIMUM 2-SAT is L-reducible to MAXIMUM NAE 3-SAT

  - $f$ transforms each clause $x \lor y$ into new clause $x \lor y \lor z$
    where $z$ is a new global variable
  - $g(C,t)$=restriction of $t$ to original variables
  - $a=1$, $b=1$
    - $z$ may be assumed false
    - each new clause is not-all-equal satisfied iff the original clause is satisfied

- **Corollary:** MAXIMUM NAE 3-SAT is not in PTAS
Inapproximability of MAXIMUM SAT(B)

- Standard reduction
  - If a variable $y$ occurs $h$ times, create new $h$ variables $y[i]$
  - Substitute $i$th occurrence with $y[i]$
  - Add ($\neg y[i] \text{ or } y[i+1]$) and ($\neg y[h] \text{ or } y[1]$)

- Not useful: deleting one new clause may increase the measure arbitrarily
  - The cycle of implications can be easily broken
  - If we add all possible implications (that is, we use a clique), then no the number of occurrences is not bounded and there is no linear relation between optimal measures

Expander graphs

- A graph $G=(V,E)$ is an expander if, for every subset $S$ of the nodes, the corresponding cut has measure at least
  \[
  \min\{|S|, |V-S|\}
  \]
  - A cycle is not an expander
  - A clique is an expander (but has unbounded degree)

- Theorem: A constant $n_0$ and an algorithm $A$ exist such that, for any $k > n_0$, $A(k)$ constructs in time polynomial in $k$ a 14-regular expander graph $E_k$ with $k$ nodes.
AP-reduction through expanders

- We may assume that $h$ is greater than $n_0$ (it suffices to replicate any clause $n_0$ times)
- For any $i$ and $j$, if $(i,j)$ is an edge of $E_h$ then add
  \((\textbf{not } y[i] \textbf{ or } y[j])\) and \((\textbf{not } y[j] \textbf{ or } y[i])\)
- Globally, we have $m + 14N$ clauses where $N$ is the sum of the $h$s
- Each variable occurs in exactly 28 new clauses and 1 old clause: hence, $B=29$
  - Starting from $B=29$, it is possible to arrive at $B=3$

Proof

- **Claim**: Any solution must satisfy all new clauses (that is, gives the same value to all copies of the same variable)
  - From the expansion property, if we change the truth value of the copies in the smaller set we do not loose anything
- $a=85$
  - $m^*(f(x)) = 14N + m^*(x) \geq 42m + m^*(x) \geq 85m^*(x)$
- $b=1$
  - $m^*(x) - m(x,t) = 14N + m^*(f(x)) - 14N - m(f(x),t) = m^*(f(x)) - m(f(x),t)$
Other inapproximability results

- **Theorem:** MINIMUM VERTEX COVER is not in PTAS
  - Reduction from MAXIMUM 3-SAT(3)

- **Theorem:** MAXIMUM CUT is not in PTAS
  - Reduction from MAXIMUM NAE 3-SAT

- **Theorem:** MINIMUM GRAPH COLORING is not in APX
  - Reduction from variation of independent set