Approximation Preserving Reductions
Summary

- Memo
- AP-reducibility
  - L-reduction technique
- Complete problems
- Examples: MAXIMUM CLIQUE, MAXIMUM INDEPENDENT SET, MAXIMUM 2-SAT, MAXIMUM NAE 3-SAT, MAXIMUM SAT($B$)
Memo: Approximation classes inclusions

- It holds that

\[ \text{PTAS} \subseteq \text{APX} \subseteq \text{log-APX} \subseteq \text{poly-APX} \subseteq \text{exp-APX} \subseteq \text{NPO} \]

where

- \( \text{log-APX} = \{ \text{polynomial-time } O(\log n)\text{-approximate problem} \} \)
- \( \text{poly-APX} = \{ \text{polynomial-time } O(n^k)\text{-approximate problem, for any } k > 0 \} \)
- \( \text{exp-APX} = \{ \text{polynomial-time } O(2^{nk})\text{-approximate problem, for any } k > 0 \} \)
Memo: Approximation classes inclusions

- Polynomial bound on $m()$ implies that any NPO problem is $h2^{nk}$-approximable for some $h$ and $k$... so every problem is in exp-APX?

- NO! There are problems for which it is even hard (NP-hard) to decide if any feasible solution exists.
  - Example: MIN $\{0,1\}$-linear programming
    - Given an integer matrix $A$ and an integer vector $b$, deciding whether a binary vector $x$ exists such that $Ax \geq b$ is NP-hard

- If $P \neq NP$, then $PTAS \subset APX \subset \log$-APX $\subset$ poly-APX $\subset$ exp-APX $\subset$ NPO
Memo: Karp reducibility

- A decision problem $P_1$ is *Karp reducible* to a decision problem $P_2$ (in short, $P_1 \leq P_2$) if there exists a polynomial-time computable function $R$ such that, for any $x$, $x$ is a YES-instance of $P_1$ if and only if $R(x)$ is a YES-instance of $P_2$

- If $P_1 \leq P_2$ and $P_2$ is in $P$, then $P_1$ is in $P$
Reducibility and NPO problems

\[ P_1 \quad \text{\textup{\textbackslash|}} \quad P_2 \]

\[ x, r \quad \rightarrow \quad f(x), r' \]

\[ r\text{-approximate solution of } x \quad \text{\textup{\textbackslash|}} \quad g(x, y) \quad \leftarrow \quad y \quad \text{\textup{\textbackslash|}} \quad r'\text{-approximate solution of } f(x) \]
A taxonomy

PTAS-reducibility → A-reducibility

P-reducibility → AP-reducibility

L-reducibility → E-reducibility → Continuous reducibility

Strict reducibility
AP-reducibility ($\leq_{AP}$)

- $P_1$ is AP-reducible to $P_2$ ($P_1 \leq_{AP} P_2$) 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 $r > 1$, $f(x, r)$ is an instance of $P_2$
  - For any instance $x$ of $P_1$, for any $r > 1$, and for any solution $y$ of $f(x, r)$, $g(x, y, r)$ is a solution of $x$
  - For any fixed $r > 1$, $f$ and $g$ are computable in polynomial time
  - For any instance $x$ of $P_1$, for any $r > 1$, and for any solution $y$ of $f(x, r)$, if $R_{P_2}(f(x, r), y) \leq r$, then $R_{P_1}(x, g(x, y, r)) \leq 1 + c(r - 1)$
Basic properties

- **Theorem:** If $P_1 \leq_{AP} P_2$ and $P_2 \in \text{APX}$, then $P_1 \in \text{APX}$

- If $A$ is an $r$-approximation algorithm for $P_2$ then
  \[ A_{P_1}(x) = g(x, A(f(x,r)), r) \]

  is a $(1+c(r-1))$-approximation algorithm for $P_1$
Basic properties

- **Theorem:** If $P_1 \leq_{AP} P_2$ and $P_2 \in \text{PTAS}$, then $P_1 \in \text{PTAS}$

- If $A$ is a polynomial-time approximation scheme for $P_2$ then
  
  $A_{P_1}(x, r) = g(x, A(f(x, r'), r'), r')$

is a polynomial-time approximation scheme for $P_1$, where

$r' = 1 + (r - 1)/c$
L-reducibility

- $P_1$ is L-reducible to $P_2$ ($P_1 \leq_L P_2$) 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 $P_1$, $m^*(f(x)) \leq a \cdot m^*(x)$
  - For any instance $x$ of $P_1$ and for any solution $y$ of $f(x)$, $|m^*(x) - m(x, g(x, y))| \leq b \cdot |m^*(f(x)) - m(f(x), y)|$
Basic property of L-reductions

- **Theorem:** If $P_1 \leq_L P_2$ and $P_2 \in \text{PTAS}$, then $P_1 \in \text{PTAS}$
  - Relative error in $P_1$ is bounded by $a \ b$ times the relative error in $P_2$

- However, in general, it is not true that
  if $P_1 \leq_L P_2$ and $P_2 \in \text{APX}$, then $P_1 \in \text{APX}$\hspace{1cm} \text{NO!}
  - The problem is that the relation between $r$ and $r'$ may be non-invertible
Basic property of L-reductions

- **Lemma:** Let $P_1$ and $P_2$ be two NPO problems such that $P_1 \leq_L P_2$.

  If $P_1 \in \text{APX}$, then $P_1 \leq_{AP} P_2$
Inapproximability of independent set

- **Theorem:** \( \text{MAX CLIQUE} \leq_{AP} \text{MAX INDEPENDENT SET} \)
  
  - \( G=(V, E), G^c=(V,V^2-E) \) is the complement graph
  - \( f(G) = G^c \)
  - \( g(G, U)=U \)
  - \( c=1 \)
    - Each clique in \( G \) is an independent set in \( G^c \)

- **Corollary:** \( \text{MAX INDEPENDENT SET} \notin \text{APX} \)
Complete problems

- AP-reduction is transitive

- AP-reduction induces a partial order among problems in the same approximation classes

- Given a class $C$ of NPO problems, a problem $P$ is $C$-hard (with respect to the AP-reduction) if, for any $P' \in C$, $P' \leq_{AP} P$. If $P \in C$, then $P$ is $C$-complete (with respect to the AP-reduction).

- For any class $C \not\subset APX$ ($\not\subset PTAS$), if $P$ is $C$-complete, then $P \not\in APX$ ($\not\in PTAS$), unless $P=NP$
MAXIMUM WEIGHTED SAT

- **INSTANCE:** CNF Boolean formula \( \varphi \) with variables \( x_1, x_2, \ldots, x_n \), with non-negative weights \( w_1, w_2, \ldots, w_n \)

- **SOLUTION:** A satisfied truth-assignment \( f \) to \( \varphi \)

- **MEASURE:** \( \max(1, \sum w_i f(x_i)) \), where \( f(x_i) = \text{true} \) is calculated as \( f(x_i) = 1 \) and \( f(x_i) = \text{false} \) as \( f(x_i) = 0 \)
NPO-complete problems

- Finding a feasible solution is as hard as SAT
  - MAX WEIGHTED SAT ∉ exp-APX
- MAX WEIGHTED SAT is NPO-complete
  - MAX WEIGHTED SAT ≤_AP MAX WEIGHTED 3-SAT
  - MAX WEIGHTED 3-SAT ≤_AP MIN WEIGHTED 3-SAT
  - MIN WEIGHTED 3-SAT ≤_AP MIN {0,1}-LINEAR PROGRAMMING
- MIN {0,1}-LINEAR PROGRAMMING is NPO-complete
APX-complete problems

- The PCP theorem permits to show that MAX 3-SAT is complete for the class of maximization problems in APX

- For any minimization problem $P$ in APX, a maximization problem $P'$ in APX exists such that $P \leq_{AP} P'$

- MAX 3-SAT is APX-complete
Inapproximability of 2-satisfiability

- **Theorem:** \( \text{MAX 3-SAT} \leq_L \text{MAX 2-SAT} \)

  - \( f \) transforms each clause \((x \text{ or } y \text{ or } z)\) into the following set of 10 clauses where \( i \) is a new variable:
    - \((x), (y), (z), (i), (\text{not } x \text{ or } \text{not } y), (\text{not } x \text{ or } \text{not } z), (\text{not } y \text{ or } \text{not } z), (x \text{ or } \text{not } i), (y \text{ or } \text{not } i), (z \text{ or } \text{not } i)\)
  
  - \( g(C,t) = \text{restriction of } t \) 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:** \( \text{MAX 2-SAT} \) is APX-complete
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 3-SAT

- **Theorem:** MAX 2-SAT $\leq_L$ MAX NAE 3-SAT
  - $f$ transforms each clause $x \text{ or } y$ into new clause $x \text{ or } y \text{ or } z$ 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:** MAX NAE 3-SAT is APX-complete
Other inapproximability results

- **Theorem:** MIN VERTEX COVER is APX-complete
  - Reduction from MAX 3-SAT(3)

- **Theorem:** MAX CUT is APX-complete
  - Reduction from MAX NAE 3-SAT

- **Theorem:** MIN GRAPH COLORING $\notin$ APX
  - Reduction from variation of independent set
The NPO world if $P \neq NP$

<table>
<thead>
<tr>
<th>NPO</th>
<th>MINIMUM TSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAXIMUM INDEPENDENT SET</td>
</tr>
<tr>
<td></td>
<td>MAXIMUM CLIQUE</td>
</tr>
<tr>
<td></td>
<td>MINIMUM GRAPH COLORING</td>
</tr>
<tr>
<td>APX</td>
<td>MINIMUM BIN PACKING</td>
</tr>
<tr>
<td></td>
<td>MAXIMUM SATISFIABILITY</td>
</tr>
<tr>
<td></td>
<td>MINIMUM VERTEX COVER</td>
</tr>
<tr>
<td></td>
<td>MAXIMUM CUT</td>
</tr>
<tr>
<td>PTAS</td>
<td>MINIMUM PARTITION</td>
</tr>
<tr>
<td>PO</td>
<td>MINIMUM PATH</td>
</tr>
</tbody>
</table>