Optimization problem: (combinatorial)
- goal: instance \(\rightarrow\) solution with min/max cost
- set of instances
- for each instance:
  - set of (valid/feasible) solutions
  - nonnegative cost of each solution \((\mathbb{R} \text{ or } \mathbb{Z})\)
- objective: min or max

\[\text{OPT}(x) = \min / \max \text{ possible cost for instance } x\]
(sometimes also the solution itself)

NP optimization problem:
- solutions have polynomial length
- instances & their solutions can be recognized in \(P\)
- cost function in \(P\)
\[\Rightarrow \text{decision problem } \in \text{NP}\]
\[\min: \text{ is } \text{OPT}(x) \leq q ? \quad (\geq \in \text{coNP})\]
\[\max: \text{ is } \text{OPT}(x) \geq q ? \quad (\leq \in \text{coNP})\]

\(NPO = \{ \text{NP optimization problems}\}\)
Approximation: \( \text{ALG} \) is a \( c \)-approximation if \( \forall x: \)

- \( \min: \quad \frac{\text{cost}(\text{ALG}(x))}{\text{cost}(\text{OPT}(x))} \leq c \quad (c \geq 1) \)
  - instance

- \( \max: \quad \frac{\text{cost}(\text{OPT}(x))}{\text{cost}(\text{ALG}(x))} \leq c \quad (c \geq 1) \)
  - e.g. 2

- \( Q:\quad \frac{\text{cost}(\text{ALG}(x))}{\text{cost}(\text{OPT}(x))} \geq c \quad (c \leq 1) \)
  - e.g. \( \frac{1}{2} \)

- usually: \( \text{ALG} \) should be polynomial time

PTAS (Polynomial-Time Approximation Scheme)

- algorithm with additional input \( \varepsilon > 0 \)
- solution is \((1+\varepsilon)\)-approximation
- polynomial time for every fixed \( \varepsilon > 0 \)
  - e.g. \( n^{2+\varepsilon} \) OK (tighter notions later)

\[ \text{PTAS} = \exists \text{NP optimization problems having PTAS} \]
\[ = \bigcap_{c>1} c-\text{APX} \]

\( \text{F-APX} = \exists \text{NP optimization problems having poly-time} \)
\[ f(n)\text{-approximation algorithm for some } f \in \text{F} \]

\[ \text{APX} = O(1)-\text{APX} \]
\[ = \text{MAX SNP in older literature} \]
\[ \text{Log-APX} = O(\log n)-\text{APX} \]
\[ \text{Poly-APX} = n^{O(1)}-\text{APX} \]

- \( P \neq \text{NP} \Rightarrow \text{PTAS} \neq \text{APX} \neq \text{Log-APX} \neq \text{Poly-APX} \) etc.
Typical approximation factors: (graph problems)
- $1 + \varepsilon$ (PTAS)
  - lots of problems on planar/H-minor-free graphs
    - e.g. H-minor-free dominating set
      - choose min. # vertices adjacent to unchosen vertices
    & in Euclidean plane e.g. TSP,
      - Steiner tree, rectilinear Steiner tree [L9]
- $\Theta(1)$ (APX-complete)
  - lots e.g. TSP, Steiner tree
  - max. coverage: choose $k$ vertices from left side of
    bipartite graph adjacent to max. # vertices
- $\Theta(\log^* n)$
  - asymmetric $k$-center: given asymmetric metric,
    choose $k$ vertices to min. max. distance $v \to$ nearest chosen
- $\Theta(\log n)$
  - set cover & dominating set
    - dominating set from left side of bipartite graph
    - max. unique coverage (exactly 1 left adjacent to right)
- \( \Theta(\log^2 n) \)
  - group Steiner tree: given graph & \( k \) groups of vertices, choose min. \# vertices inducing connected subgraph & containing at least 1 vertex in each group
  
- \( \Omega(\log^2 n) \land O(n^\varepsilon) \) (OPEN)
  - directed Steiner tree: given graph, \( k \) terminal vertices, & root vertex, choose min. \# vertices inducing root-to-terminal path for each terminal
  
- \( \Omega(\log^{1-\varepsilon} n) \land O(n^c) \) (OPEN)
  
  \( c = \frac{1}{3} \) &\Rightarrow label \ cover \ (MinRep \ & \ MaxRep) \ [future \ lecture] 
  
  \( c = \frac{4}{5} + \varepsilon \) &\Rightarrow directed \ Steiner \ forest: \ given \ \( s_i \rightarrow t_i \) \ pairs, choose min. \# vertices inducing such paths
  
- \( \Omega(n^{1-\varepsilon}) \land O(n) \) (\( \uparrow \) polylog factors)

- chromatic number: min \( k \) such that \( k \)-colorable
- independent set = clique (complement graph)
Approximation preserving reductions: $A \Rightarrow B$

(see Crescenzi - CCC 1997)

$A$ instance $x$ \xrightarrow{f} $B$ instance $x' = f(x)$

$A$ solution $y = g(x, y')$ to $x$ \xleftarrow{g} $B$ solution $y'$ to $x'$

PTAS-reduction: $\forall \varepsilon > 0 \exists S = S(\varepsilon) > 0$ such that $y'$ is $(1 + S(\varepsilon))$-approximation to $B$ \Rightarrow $y = g(x, y')$ is $(1 + \varepsilon)$-approximation to $A$

[Crescenzi & Trevisan 1994]

- $f$ & $g$ can depend on $\varepsilon$ too (else "P-reduction")
- $B \in$ PTAS $\Rightarrow A \in$ PTAS
- $A \in$ PTAS $\Rightarrow B \in$ PTAS
- ditto for APX
- careful: $A \in$ PTAS $\Rightarrow B \notin$ PTAS
- if $S(\emptyset) = \emptyset$ also works then also NP reduction
- reductions chain: $A \Rightarrow B \Rightarrow C$

AP-reduction: $S(\varepsilon) = O(\varepsilon)$

- $B \in O(f)$-APX $\Rightarrow A \in O(f)$-APX

Strict reduction: $S(\varepsilon) = \varepsilon$

[Orponen & Mannila 1987]
\[ \text{APX-hard} = \exists \text{ PTAS-reduction from any problem } \in \text{APX} \]
\[ \quad \& \not\in \text{PTAS} \quad \text{if } P \neq \text{NP} \]

\[ \overline{\text{O}(\varepsilon)\text{-APX-hard}} = \exists \text{ AP-reduction from any problem } \in \text{O}(\varepsilon)\text{-APX} \]
\[ \quad \text{(other definitions possible)} \]
\[ \quad \& \not\in o(\varepsilon)\text{-APX} \quad \text{if } P \neq \text{NP} \]

L-reduction:
\[ \text{OPT}_B (x') = O(\text{OPT}_A (x)) \]
\[ \quad |\text{cost}_A (y) - \text{OPT}_A (x)| = O(|\text{cost}_B (y') - \text{OPT}_B (x')|) \]
\[ \quad \leq \beta \]
[\text{Papadimitriou \\ Yannakakis - JCSS 1991}]

\[ \Rightarrow \text{PTAS-reduction} \]

\[ \Rightarrow \text{AP-reduction} \text{ with } S(\varepsilon) = \varepsilon / \alpha \beta : \]

\[ \frac{\text{cost}_A (y)}{\text{OPT}_A (x)} \leq \frac{\text{OPT}_A (x) + \beta (\text{cost}_B (y') - \text{OPT}_B (x'))}{\text{OPT}_A (x)} \]
\[ \leq 1 + \alpha \beta \left( \frac{\text{cost}_B (y') - \text{OPT}_B (x')}{\text{OPT}_B (x')} \right) \]
\[ = 1 + \alpha \beta \left( \frac{\text{cost}_B (y')}{\text{OPT}_B (x')} - 1 \right) \]
\[ \leq 1 + S(\varepsilon) = 1 + \varepsilon / \alpha \beta \]
\[ \leq 1 + \varepsilon . \quad \square \]

- also \text{NP reduction}
- most popular reduction type
\[ \text{L-reduction } \rightarrow \text{ PTAS-reduction, max case: (uncovered)} \]

\[
\begin{align*}
\cosct_A(y) &= \text{OPT}_A(x) - \left(\text{OPT}_A(x) - \cosct_A(y)\right) \\
&\leq \text{OPT}_B(x) - \cosct_B(y) \\
&\geq \text{OPT}_A(x) - \beta (\text{OPT}_B(x') - \cosct_B(y')) \\
\frac{\cosct_A(y)}{\text{OPT}_A(x)} &\geq \frac{\text{OPT}_A(x) - \beta (\text{OPT}_B(x') - \cosct_B(y'))}{\text{OPT}_A(x)} \\
&= 1 - \beta \frac{\text{OPT}_B(x') - \cosct_B(y')}{\text{OPT}_A(x)} \\
&\geq 1 - \alpha \beta \frac{\text{OPT}_B(x') - \cosct_B(y')}{\text{OPT}_B(x')} \\
&= 1 - \alpha \beta \left(1 - \frac{\cosct_B(y')}{\text{OPT}_B(x')}\right) \\
&\geq \frac{1}{1+\delta}
\end{align*}
\]

\[
\geq 1 - \alpha \beta + \frac{\alpha \beta}{1+\delta} = \frac{1}{1+\delta}
\]

when \( \delta = \frac{1}{\alpha \beta (1+\frac{1}{\epsilon})-1} = \frac{\epsilon}{\alpha \beta (1+\frac{1}{\epsilon})-1} \)

\[
1+\delta = \frac{\alpha \beta (1+\frac{1}{\epsilon})}{\alpha \beta (1+\frac{1}{\epsilon})-1} \\
\frac{1}{1+\delta} = \frac{\alpha \beta (1+\frac{1}{\epsilon})-1}{\alpha \beta (1+\frac{1}{\epsilon})}
\]

\[
\frac{\alpha \beta}{1+\delta} + 1 = \frac{\alpha \beta (1+\frac{1}{\epsilon})+\frac{1}{\epsilon}}{1+\frac{1}{\epsilon}} \\
\frac{\alpha \beta}{1+\delta} + 1 - \alpha \beta = \frac{\frac{1}{\epsilon}}{1+\frac{1}{\epsilon}} = \frac{1}{\epsilon+1}
\]

\[\text{CLEANER: } y' \text{ is a } (1-\frac{\epsilon}{\alpha \beta})-\text{approximation} \quad (c<1 \text{ view})\]

\[\Rightarrow y \text{ is a } (1-\epsilon)-\text{approximation} \quad [\text{Williamson & Shmoys book, 2010}]\]
APX-complete problems:

Max E3SAT-E5: exactly 3 distinct literals/clause & exactly 5 occurrences/variable
[Feige - J.ACM 1998]

Max 3SAT-3: [Papadimitriou & Yannakakis - JCSS 1991]
- usual 3SAT ⇒ 3SAT-3 reduction:

- not approximation preserving: can now set variable \( x \) half true & half false at cost of one violation (can't bound damage)
- fix: connect copies \( x_1, x_2, \ldots, x_k \) with an expander graph where edge is \( x_i = x_j (\overline{x_i} \lor x_j) \)
  - bounded degree, \( k \) nodes
  - \( \forall \text{cut } (A,B): \# \text{cross edges } \geq \min |A|, |B| \)
  (simplification of PY91 construction by Crescenzi 1997)
- setting \( x_i \)'s to majority value won't decrease \# satisfied clauses

⇒ 3SAT-\( O(1) \) is APX-hard

\( \rightarrow 29 = 2.14 + 1 \) using 14-regular expander
[Lubotzky, Phillips, Sarnak - Combinatorica 1988]
- then use usual reduction ⇒ 3SAT-3

\( O(k) \) violations ⇔ \( k \) violations

⇒ L-reduction actually a gap reduction, not L-reduction
Independent set, max. degree $\Delta = O(1)$
- any maximal independent set is $\Delta$-approximation
- strict-reduction from Max 3SAT-3
  \[\text{[Papadimitriou & Yannakakis - JCSS 1991]}\]
- variable gadget $\Rightarrow$ indep. set can't use $x_i$ & $\overline{x_i}$
- clause gadget $\Rightarrow \leq 1$ point, 0 if not satisfied
- max. degree 4
- 3-regular also APX-complete \[\text{[Berman & Fujito - TCS 1999]}\]
  & \[\text{[Alimonti & Kann - TCS 2000]}\]

Vertex cover
- greedy algorithm is 2-approximation
- L-reduction from Independent set: do nothing
  \[\text{[Papadimitriou & Yannakakis - JCSS 1991]}\]
- vertex cover $\Leftrightarrow$ complement is independent
- $OPT_{vc}$ & $OPT_{is}$ both $\Theta(|V|)$
  for bounded-degree graphs $\Rightarrow \Theta$(each other)
- absolute error preserved
- 3-regular OK

Dominating set, max. degree $\Delta = O(1)$
- any minimal dominating set is $\Delta$-approximation
- strict-reduction from Vertex cover:
  \[\text{[Papadimitriou & Yannakakis - JCSS 1991]}\]

\[\text{v} \quad \leftrightarrow \quad \text{v} \quad \Rightarrow \quad \text{v} \quad \text{w} \quad \Rightarrow \quad \text{v} \quad \text{v} \quad \text{w}\]

\[\Rightarrow \text{never need to choose edge node (move to v)}\]
- 3-regular OK