6.890	Lecture 10	Oct. 7, 2014

Optimization problem: (combinatorial)

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

OPT(x) = min/max 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 $\in P$

⇒ decision problem $\in NP$
↳ min: is $OPT(x) \le q$? ($\ge \in coNP$)
max: is $OPT(x) \ge q$? ($\le \in coNP$)

NPO = {NP optimization problems}

Approximation: ALG is a c-approximation if $\forall x$:
$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad$ ↓ instance

- min: $\dfrac{\text{cost}(\text{ALG}(x))}{\text{cost}(\text{OPT}(x))} \le c$ $\quad$ $(c \geq 1)$
- max: $\dfrac{\text{cost}(\text{OPT}(x))}{\text{cost}(\text{ALG}(x))} \le c$ $\quad$ $(c \geq 1)$ e.g. 2

  OR: $\dfrac{\text{cost}(\text{ALG}(x))}{\text{cost}(\text{OPT}(x))} \geq c$ $\quad$ $(c \le 1)$ e.g. $\frac{1}{2}$
- usually: 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. $n^{2^{1/\varepsilon}}$ OK $\quad$ (tighter notions later)

PTAS = {NP optimization problems having PTAS}

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

APX $= O(1)$-APX
$\quad$ = MAX SNP in older literature

Log-APX $= O(\lg n)$-APX

Poly-APX $= n^{O(1)}$-APX

- $P \neq NP \Rightarrow$ PTAS $\subsetneq$ APX $\subsetneq$ Log-APX $\subsetneq$ 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

"dual"

- $\Theta(\log^* n)$
  - asymmetric k-center: given asymmetric metric, choose k vertices to min. max distance v → 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
- $\Omega(\log^2 n) \cap O(n^\varepsilon)$ (OPEN)
  - directed Steiner tree: given graph, k terminal vertices, & root vertex, choose min. # vertices inducing root-to-terminal path for each terminal
- $\Omega(2^{\log^{1-\varepsilon} n}) \cap O(n^c)$ (OPEN)
  - $c=\frac{1}{3}$ → label cover (MinRep & MaxRep) [future lecture]
  - $c=\frac{4}{5}+\varepsilon$ → directed Steiner forest: given $s_i \to t_i$ pairs, choose min. # vertices inducing such paths
- $\Omega(n^{1-\varepsilon}) \cap \tilde{O}(n)$
  - ↑ polylog factors
  - chromatic number: min k such that k-colorable
  - independent set ≡ clique (complement graph)

Approximation preserving reductions: $A \to B$

(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: $\forall \varepsilon > 0 \ \exists \delta = \delta(\varepsilon) > 0$ such that

y' is $(1+\delta(\varepsilon))$-approximation to B

$\Rightarrow y = g(x, y')$ is $(1+\varepsilon)$-approximation to A

[Crescenzi & Trevisan 1994]

- f & g can depend on ε too (else "P-reduction")
- $B \in PTAS \Rightarrow A \in PTAS$ (chain algs. together)
- $A \notin PTAS \Rightarrow B \notin PTAS$
- ditto for APX
- careful: $A \in PTAS \not\Rightarrow B \in PTAS$
- if $\delta(\emptyset) = \emptyset$ also works then also NP reduction
- reductions chain: $A \to B \to C$

AP-reduction: $\delta(\varepsilon) = O(\varepsilon)$ ($\leq c \cdot \varepsilon$, no growth assumption)

[Crescenzi, Kann, Silvestri, Trevisan 1995]

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

Strict reduction: $\delta(\varepsilon) = \varepsilon$ [Orponen & Mannila 1987]

A-reduction: y' is c-approx. $\Rightarrow y$ is $O(c)$-approx.

<u>APX-hard</u> $= \exists$ PTAS-reduction from any problem $\in$ APX
- $\notin$ PTAS if $P \neq NP$

<u>$O(f)$-APX-hard</u> $= \exists$ A-reduction from any problem $\in O(f)$-APX
(other definitions possible)
- $\notin o(f)$-APX if $P \neq NP$

<u>L-reduction</u>: $\mathrm{OPT}_B(x') = \overset{\nearrow \le \alpha \cdot}{O}(\mathrm{OPT}_A(x))$
& $|\mathrm{cost}_A(y) - \mathrm{OPT}_A(x)| = \underset{\hookrightarrow \le \beta \cdot}{O}(|\mathrm{cost}_B(y') - \mathrm{OPT}_B(x')|)$
[Papadimitriou & Yannakakis — JCSS 1991]

$\Rightarrow$ PTAS-reduction
- for minimization problems:
  $\Rightarrow$ AP-reduction with $\delta(\varepsilon) = \varepsilon / \alpha\beta$:

$$\frac{\mathrm{cost}_A(y)}{\mathrm{OPT}_A(x)} \le \frac{\mathrm{OPT}_A(x) + \beta(\mathrm{cost}_B(y') - \mathrm{OPT}_B(x'))}{\mathrm{OPT}_A(x)}$$

$$\le 1 + \alpha\beta\left(\frac{\mathrm{cost}_B(y') - \mathrm{OPT}_B(x')}{\mathrm{OPT}_B(x')}\right)$$

$$= 1 + \alpha\beta\Big(\underbrace{\frac{\mathrm{cost}_B(y')}{\mathrm{OPT}_B(x')}}_{\le 1 + \delta(\varepsilon) = 1 + \varepsilon/\alpha\beta} - 1\Big)$$

$$\le 1 + \varepsilon. \quad \square$$

- also NP reduction
- most popular reduction type

L-reduction → PTAS-reduction, max case: (uncovered)

$$\mathrm{cost}_A(y) = \mathrm{OPT}_A(x) - \underbrace{(\mathrm{OPT}_A(x) - \mathrm{cost}_A(y))}_{\le \beta\cdot(\mathrm{OPT}_B(x') - \mathrm{cost}_B(y'))}$$

$$\ge \mathrm{OPT}_A(x) - \beta(\mathrm{OPT}_B(x') - \mathrm{cost}_B(y'))$$

$$\frac{\mathrm{cost}_A(y)}{\mathrm{OPT}_A(x)} \ge \frac{\mathrm{OPT}_A(x) - \beta(\mathrm{OPT}_B(x') - \mathrm{cost}_B(y'))}{\mathrm{OPT}_A(x)}$$

$$= 1 - \beta\,\frac{\mathrm{OPT}_B(x') - \mathrm{cost}_B(y')}{\mathrm{OPT}_A(x)}$$

$\mathrm{OPT}_B(x') \le \alpha\cdot \mathrm{OPT}_A(x)$

$\mathrm{OPT}_A(x) \ge \frac{1}{\alpha}\mathrm{OPT}_B(x')$

$\frac{1}{\mathrm{OPT}_A(x)} \le \alpha\,\frac{1}{\mathrm{OPT}_B(x')}$

$-\frac{1}{\mathrm{OPT}_A(x)} \ge -\alpha\,\frac{1}{\mathrm{OPT}_B(x')}$

$$\ge 1 - \alpha\beta\,\frac{\mathrm{OPT}_B(x') - \mathrm{cost}_B(y')}{\mathrm{OPT}_B(x')}$$

$$= 1 - \alpha\beta\Big(1 - \underbrace{\frac{\mathrm{cost}_B(y')}{\mathrm{OPT}_B(x')}}_{\ge \frac{1}{1+\delta}}\Big)$$

$$\ge 1 - \alpha\beta + \frac{\alpha\beta}{1+\delta}$$

$$= \frac{1}{1+\varepsilon} \text{ when } \delta = \frac{1}{\alpha\beta(1+\frac{1}{\varepsilon}) - 1} = \frac{\varepsilon}{\alpha\beta}\Big/\Big(1+\varepsilon - \frac{\varepsilon}{\alpha\beta}\Big)$$

$$1+\delta = \frac{\alpha\beta(1+\frac{1}{\varepsilon})}{\alpha\beta(1+\frac{1}{\varepsilon})-1} \qquad \frac{1}{1+\delta} = \frac{\alpha\beta(1+\frac{1}{\varepsilon})-1}{\alpha\beta(1+\frac{1}{\varepsilon})}$$

$$\frac{\alpha\beta}{1+\delta}+1 = \frac{\alpha\beta(1+\frac{1}{\varepsilon})+\frac{1}{\varepsilon}}{1+\frac{1}{\varepsilon}} \qquad \frac{\alpha\beta}{1+\delta}+1-\alpha\beta = \frac{\frac{1}{\varepsilon}}{1+\frac{1}{\varepsilon}} = \frac{1}{\varepsilon+1}$$

CLEANER: y' is a $(1-\varepsilon/\alpha\beta)$-approximation
$\Rightarrow y$ is a $(1-\varepsilon)$-approximation (c < 1 view)

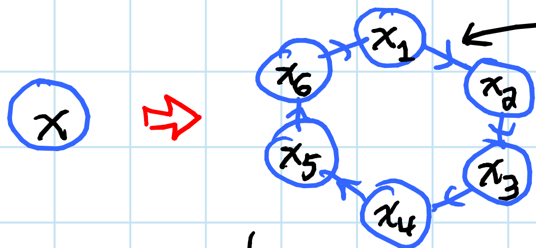
[Williamson & Shmoys book, 2010]

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:



$\neg x_1 \vee x_2$

i.e. $x_1 \Rightarrow x_2$

  - not approximation preserving: can now set variable x half true & half false at cost of one violation (can't bound damage)

- fix: connect copies $x_1, x_2, \dots, x_k$ with an expander graph where edge is $x_i = x_j$ $\begin{pmatrix} \neg x_i \vee x_j \\ \neg x_j \vee x_i \end{pmatrix}$
  - ↳ bounded degree, k nodes
  - ↳ $\forall$ cut (A, B): # 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

↳ $29 = 2 \cdot 14 + 1$ using 14-regular expander [Lubotzky, Phillips, Sarnak – Combinatorica 1988]

- then use usual reduction → 3SAT-3

  $O(k)$ violations ⇐ k violations

⇒ L-reduction

Independent set, max. degree $\Delta = O(1)$

- any maximal indep. set is Δ-approximation
- strict-reduction from Max 3SAT-3

  [Papadimitriou & Yannakakis – JCSS 1991]

  - variable gadget $\Rightarrow$ indep. set can't use x_i & $\bar{x}_i$
  - clause gadget $\Rightarrow \leq 1$ point, 0 if not satisfied
  - max. degree 4
- 3-regular also APX-complete [Berman & Fujito – TCS 1999] & [Alimonti & Kann – TCS 2000]

Vertex cover

- greedy algorithm is 2-approximation
- L-reduction from Independent set: do nothing

  [Papadimitriou & Yannakakis – JCSS 1991]

  - vertex cover $\Leftrightarrow$ complement is independent
  - OPT_{VC} & OPT_{IS} both $\Theta(|V|)$ for bounded-degree graphs $\Rightarrow \Theta(\text{each other})$
  - absolute error preserved
- 3-regular OK

Dominating set, max. degree $\Delta = O(1)$

- any mimimal dominating set is Δ-approximation
- strict-reduction from Vertex cover:

  [Papadimitriou & Yannakakis – JCSS 1991]

  $\Rightarrow$ never need to choose edge node (move $\to v$)
- 3-regular OK

6.890 Algorithmic Lower Bounds: Fun with Hardness Proofs
Fall 2014