Announcements:

- final
 - thursday, 3-5pm.
 - cumulative
 - 1 page handwritten cheat-sheet

Last time:

- pseudo polynomial time
- Knapsack PTAS

Today:

- online algorithms
- ski renter
- secretary

Approximation Algorithms

"show algorithm's solution is always close to optimal solution"

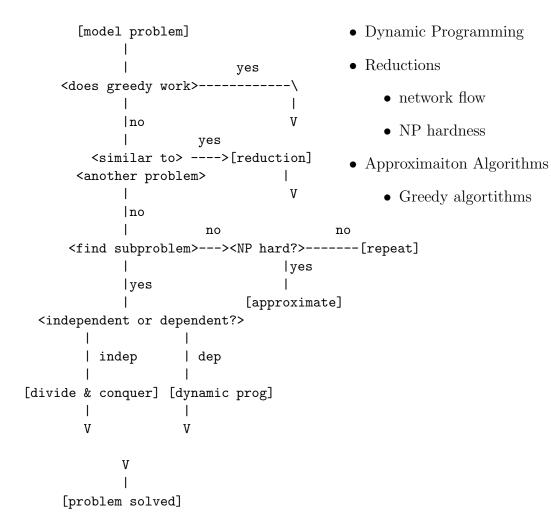
Challenge: for hard problems optimal solution is complex.

Approach:

- 1. relax constraints and solve relaxed optimally.
- 2. fix violated constraints.
- 3. show "fixed solution" is close to "relaxed solution"

Algorithms Flow Chart

Course Topics



Online Algorithms

"algorithms that must make decisions without full knowledge of input"

(e.g., if input is events over time, then algorithm doesn't know future)

Ski Renter

input:

- cost to buy skis: B.
- cost to rent skis: R.
- daily weather d_1, \dots, d_n with $d_i = \begin{cases} 1 & \text{if good weather} \\ 0 & \text{if bad weather} \end{cases}$ (let $k = \sum_i d_i$)

output: schedule for renting or buying skis.

online constraint: on day i do not know d_{i+1}, \ldots, d_n .

Note: optimality is impossible because don't know future.

Idea: approximate "optimal offline" algorithm

Algorithm: OPT (offline)

- if kR < B, buy on day 1.
- $\bullet\,$ else, rent on each good day.

Performance: $OPT = \min(kR, B)$.

Def: an online alg is β -competitive with optimal offline alg, OPT, if on all inputs x for X,

• minimization: $Alg(x) \leq \beta OPT(x)$.

• maximization: $Alg(x) \ge OPT(x)/\beta$.

Challenge:

- if we buy first day we ski:
 - for $d = (1, 0, 0, \dots, 0)$
 - OPT = R; $Alq = B \gg R$
- if we rent each time we ski
 - for $d = (1, 1, 1, \dots, 1)$
 - OPT = B; $Alq = Rn \gg B$

Algorithm: "Rent to Buy" "Rent unless total rental cost would exceed buy cost, then buy"

Example: R = 1, B = 3

$$Alg = \underbrace{3R + B}_{\leq 2B}, OPT = B$$

Theorem: $Alg \leq 2OPT$ (Alg is 2-competitive)

Proof:

case 1: $kR \le B$

- Alg: kR
- OPT: kR

$$\Rightarrow Alg = OPT \le 2OPT.$$

case 2: kR > B

- Alg: total rental $+ B \le 2B$
- OPT: B

$$\Rightarrow Alg \leq 2OPT$$
.

Note: competitive analysis gives very strong approximation result.

Secretary Problem

input:

- sequence of candidates $1, \ldots, n$.
- ordering on candidate qualities.

output:

- "hire" / "no hire" decisions.
- to hire best candidate.

online constraint: must make hire/no hire decision for i before seeing i + 1, ..., n.

Fact: "optimal offline" always hires best secretary.

Claim: no deterministic algorithm approximates optimal offline.

Proof: two candidates

case 1: Alg hires 1

• 2 is better.

case 2: Alg doesn't hire 1

• 1 is better.

Idea: consider randomized algorithms.

(maximize probability of hiring the best candidate.)

Claim: randomized algorithm is *n*-competitive offline.

Proof:

- Alg: for all i, pick ith secretary with probability 1/n.
- ullet Alg is right with probability 1/n
- OPT is always right.

 \Rightarrow *n*-competitive.

Claim: no algorithm hires best candidate with probability $\Omega(1/n)$.

Idea: consider randomized inputs.

Assumption: candidates arrive in a uniformly random order.

Example: n = 3

Two algs for example:

(a) take i candidate for some i

$$\Rightarrow \mathbf{Pr}[\text{success}] = 1/3$$

- (b) look at 1st, condition choice of 2nd or 3rd.
 - if 2nd better than 1st, hire 2nd
 - else, hire 3rd.

$$\Rightarrow \mathbf{Pr}[\text{success}] = 1/2$$

Algorithm: Secretary Alg

- \bullet interview k candidates but make no offers
- hire next secretary that is better than any of first k.

Lemma: For k = n/2 alg is 4-competitive.

Proof:

- hire best when 2nd best in first half and 1st best in second half.
- Recall: $\mathbf{Pr}[A\&B] = \mathbf{Pr}[A \mid B] \mathbf{Pr}[B]$.
- $\mathbf{Pr}[2nd \text{ best in first half}] = 1/2$
- **Pr**[1st best in second half | 2nd best in first half] = $\frac{n/2}{n-1} \ge 1/2$

 $\Rightarrow \begin{array}{l} \mathbf{Pr}[\text{hire best}] & \geq \\ \mathbf{Pr}[\text{2nd in 1st } 1/2] \, \mathbf{Pr}[\text{1st in 2nd } 1/2 \mid \text{2nd in 1st } 1/2] \geq \\ 1/4. \end{array}$

Question: what is best k?

Theorem: for k = 1/e alg is e-competitive

and this is best possible.