Dynamic Programming Lecture #1

Outline:

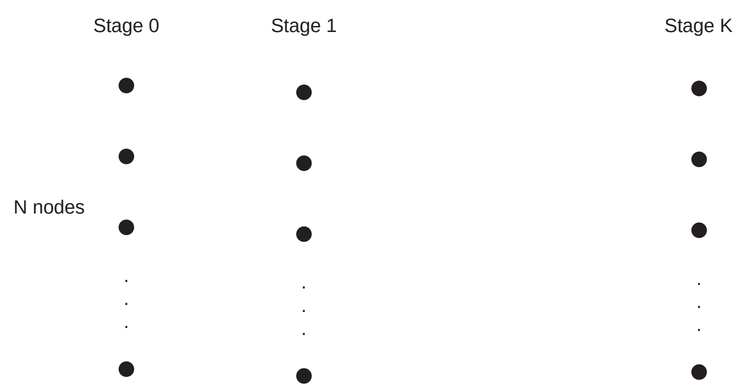
- Problem formulation(s)
- Principle of optimality
- Issues and variations

Motivation: Staged Optimization

- Q: How to formulate optimization for problems that occur in "stages"?
- "Standard" Optimization:

$$\min_{\theta \in \Theta} J(\theta)$$

  - Cost function: $J(\theta) = J(\theta_1, \ldots, \theta_n)$
  - Variables: $(\theta_1, \ldots, \theta_n) = \theta$
  - Constraints: $\theta \in \Theta$
- Examples: Best fit of experimental data, variation of design parameters, etc.
- Example: Shortest Path

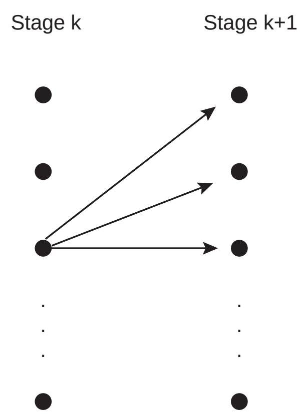


  - Setup: Hop from one node to next
  - Define: $\ell_{ij}^k \stackrel{\text{def}}{=}$ distance from node i to j at stage k
  - Objective: Minimize

$$\sum_{k=0}^{K-1} \ell_{ij}^k$$

  - Tradeoff: Immediate distance versus future distances

Nondeterministic Path Evolution



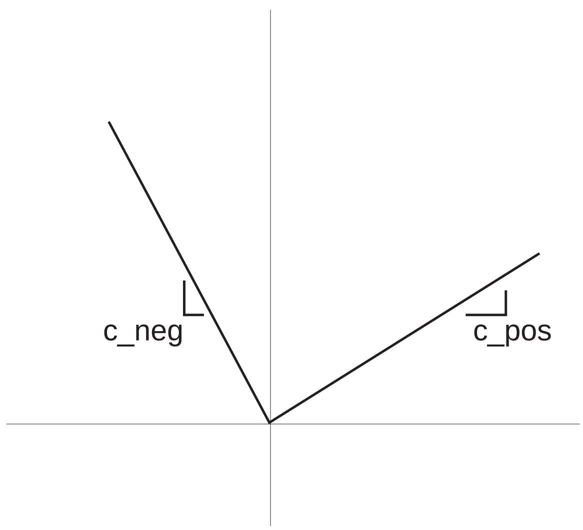
- Setup: Hop from one node to next, *but*
  ACTUAL DESTINATION = DESIRED DESTINATION $+1$, $+0$, -1
  i.e., uncertain evolution
- Consequences
  - Cost function $\sum_{k=0}^{K-1} \ell_{ij}^k$ not fully specified
  - Must specify "contingency" rules
  - Must model nondeterminism

Example: Inventory Control

- Inventory model:

$$x^+ = x + u - D$$

tomorrow's inventory = today's inventory + production - demand



- Cost:

$$\sum_{k=0}^{N} c_{\text{pos}} x_{\text{pos}} + c_{\text{neg}} x_{\text{neg}}$$

  - $x > 0 \Rightarrow$ storage cost
  - $x < 0 \Rightarrow$ backlog cost
- Demand $d \in \{0, d_{\text{low}}, d_{\text{high}}\}$
- Decisions:
  - How much to produce?
  - How to model demand?

Modeling Nondeterminism

- Let π denote a "policy", i.e., a set of contingency rules
- Let w denote nondeterministic elements
- Overall cost is a function of *both*:

$$J(\pi, w)$$

- How to model w?
  - Random:

$$\min_{\pi} \mathrm{E}_w \left[J(\pi, w) \right]$$

  - Worst case:

$$\min_{\pi} \max_{w} J(\pi, w)$$

  - Risk sensitive: $0 < \alpha < \infty$

$$\min_{\pi} \mathrm{E}_w \left[e^{\alpha J(\pi, w)} \right]$$

  - Game theoretic: w penalized according to $G(\pi, w)$ (is $G(\cdot)$ known?)
- Examples
  - Series of coin tosses: $\{T, T, T, H, T, H, T, T, ...\}$
  - Payoff: $2^{\text{first occurrence of } H}$
  - "Expected" payoff with fair coin:

$$(1/2) \cdot 2 + (1/4) \cdot 2^2 + (1/8) \cdot 2^3 + ... = \infty$$

  - Risk sensitive reward with fair coin: $\log(\text{payoff})$ (discounts large rewards)

$$(1/2) \log(2) + (1/4) \log(2^2) + (1/8) \log(2^3) + ... < \infty$$

  - Worst case payoff $= 1$
  - Game theoretic payoff?
- How to *model* w in inventory control?

Example: Asset Management

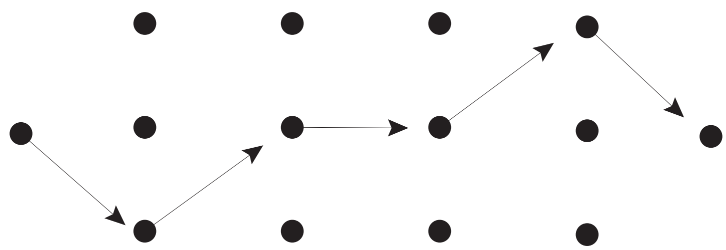
- Have property...get buy offer: w_k
- Do we sell? hold?
- Costs:
  - If we hold, we must pay to keep on market
  - If we hold until end, we must accept final offer
  - If we sell, we may miss future offers
- Model of offers: $w_k \in \{w_{\text{low}}, w_{\text{mid}}, w_{\text{high}}\}$ with probabilities.
- Similar to "parking lot" dilemma

Example: Hypothesis Testing

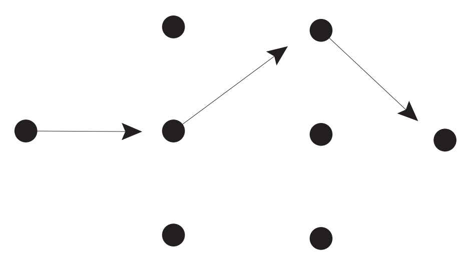
- Gambling game involving opponent with dice.
- Two possibilities:
  1. Rolling fair dice
  2. Rolling crooked dice
- Q: Is opponent cheating?
- Costs:
  - If we make correct conclusion, we are rewarded
  - If we make incorrect conclusion, we are penalized
  - If we continue to play, we are penalized
- Common theme: Distinction from “standard” optimization
  - Staged evolution
  - Uncertain evolution

Principle of Optimality

Original Problem Optimal Path



Subproblem Optimal Path?



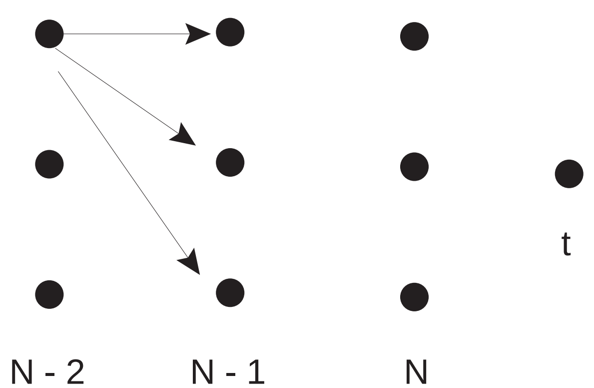
- Optimal course for subproblem = tail of optimal original problem
- Why? If not, then original course can be improved
- Utility: Reduction in computations
- Introduce "terminal node" t
- Define $J_k(i)$ = minimum distance from node i to t starting at stage k
- Clearly

$$J_N(1) = \ell_{1t}^N$$
$$J_N(2) = \ell_{2t}^N$$
$$J_N(3) = \ell_{3t}^N$$

(no choice)

Principle of Optimality, cont (2)

- How to compute $J_{N-1}(1)$? Compare...
  - $\ell_{11}^{N-1} + J_N(1)$
  - $\ell_{12}^{N-1} + J_N(2)$
  - $\ell_{13}^{N-1} + J_N(3)$
- $J_{N-1}(1)$ is smallest of 3 choices
- How to compute $J_{N-2}(1)$?



- Total # of paths = 9...but only need to check 3!
  - $\ell_{11}^{N-2} + J_{N-1}(1)$
  - $\ell_{12}^{N-2} + J_{N-1}(2)$
  - $\ell_{13}^{N-3} + J_{N-1}(3)$

Principle of Optimality, cont (3)

- Can proceed backward to compute $J_{N-3}(i), \ldots, J_1(i)$
- Minimum total cost from start node s? Compare...
  - $\ell_{s1} + J_1(1)$
  - $\ell_{s2} + J_1(2)$
  - $\ell_{s3} + J_1(3)$
- Compare: N stages & m nodes $= m^N$ # paths
- Using DP: # comparisons-per-stage $= m^2 \Rightarrow Nm^2$ total comparisons
- Principle of optimality DISQUALIFIES all but optimal "tails"

Issues & Variations

- Generalization: Main idea is "simply" principle of optimality
- Key question: How to REPRESENT different optimization to fit DP framework.
- Random element: Presence of stochastic/random behaviors in evolution of stages must be modeled.
- Information: What is optimal policy given limited information about current situation?
- Horizon: What if there is no clear "termination" stage?

$$\sum_{k=0}^{\infty} e^{-\alpha k} h(x_k) \quad \text{vs} \quad \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} h(x_k)$$

- Curse of dimensionality: Dynamic programming reduces search...but still can be huge.