# Dynamic Programming

Taek-Ho Lee

Department of Industrial Engineering, SeoulTech

Mail: taekho.lee@seoultech.ac.kr

# Introduction

- Dynamic Programming
  - A mathematical technique useful for a sequence of interrelated decisions, aimed at deriving the optimal combination of decisions.
  - It does not have a standard mathematical formulation.

# Prototype Example

- Subway Transfer Problem
  - When traveling from Station A to destination Station J, multiple transfer routes are possible.
  - Multiple transfer routes
    - ✓ Each segment between two stations has a known travel time.
    - ✓ The passenger wants to minimize the total travel time.
  - The journey from A to J requires several transfers, but simply choosing the shortest segment time at each step does not necessarily lead to the overall minimum travel time.

# Prototype Example

- Subway Transfer Problem
  - Greedy choice (locally optimal choice at each step)
    - ✓ If, in this problem, the passenger always chooses the next transfer with the shortest travel time, the path selected would be A-B-F-I-J, with a total travel time of 13.
    - ✓ However, if the route is adjusted at an intermediate stage, for example A-D-F-I-J, the total travel time becomes 11.

| | B | C | D |
|---|---|---|---|
| A | 2 | 4 | 3 |

| | E | F | G |
|---|---|---|---|
| B | 7 | 4 | 6 |
| C | 3 | 2 | 4 |
| D | 4 | 1 | 5 |

| | H | I |
|---|---|---|
| E | 1 | 4 |
| F | 6 | 3 |
| G | 3 | 3 |

| | J |
|---|---|
| H | 3 |
| I | 4 |

# Prototype Example

- Subway Transfer Problem
  - Mathematical Formulation
    - ✓ $x_n$: the next transfer station chosen at the n-th move.
    - ✓ Given the current station $s$, the remaining total travel time when passing through $x_n$ is $f_n(s, x_n)$
      - $f_n(s, x_n) = c_{sx_n} + f^*_{n+1}(x_n)$
        - $f^*_{n+1}(x_n)$ is the minimum travel time from station $x_n$ at step n+1 to the final destination (assuming optimal choices thereafter)
      - $f^*_n(s) = \min_{x_n} f_n(s, x_n) = f_n(s, x^*_n)$
        - $x^*_n$ is the decision that minimizes $f_n(s, x_n)$, i.e., the remaining total travel time from the current station $s$ to the destination via the next station.
    - ✓ The objective of the overall problem is to find $f^*_1(A)$, the minimum total travel time starting from the first station A and the corresponding sequence of transfer stations.

# Prototype Example

- Subway Transfer Problem
  - Solution procedure
    - ✓ For $n = 4$, the current station $s$ can only be H or I, and in each case the only possible choice is $x_4 = J$.
    - ✓ If we make a table for the current station, the minimum remaining travel time, and the corresponding current choice, it becomes:

| $s$ | $f_4^*(s)$ | $x_4^*$ |
|---|---|---|
| H | 3 | J |
| I | 4 | J |

# Prototype Example

- Subway Transfer Problem
  - Solution procedure
    - ✓ For $n = 4$, the current station $s$ can only be H or I, and in each case $x_4 = J$.
    - ✓ The minimum remaining travel time, and the corresponding current choice:

| $s$ | $f_4^*(s)$ | $x_4^*$ |
|---|---|---|
| H | 3 | J |
| I | 4 | J |

    - ✓ For $n = 3$, the possible current $s$ are E, F, G. The next is H or I, selecting the better option.
      - We can summarize in a table:

| $s$ | $f_3(s, x_3) = c_{sx_3} + f_4^*(x_3)$ | | $f_3^*(s)$ | $x_3^*$ |
|---|---|---|---|---|
| | H | I | | |
| E | 4 | 8 | 4 | H |
| F | 9 | 7 | 7 | I |
| G | 6 | 7 | 6 | H |

# Prototype Example

- Subway Transfer Problem
  - Solution procedure
    - ✓ For $n = 2$, the current station $s$ can be B, C, D. The next choice is E, F, or G, selecting the better option.

| $s$ | $f_2(s, x_2) = c_{sx_2} + f_3^*(x_2)$ | | | $f_2^*(s)$ | $x_2^*$ |
|---|---|---|---|---|---|
| | E | F | G | | |
| B | 11 | 11 | 12 | 11 | E or F |
| C | 7 | 9 | 10 | 7 | E |
| D | 8 | 8 | 11 | 8 | E or F |

    - ✓ For $n = 1$, the current station $s$ can only be A. The next choice is B, C, or D, selecting the better option.

| $s$ | $f_1(s, x_1) = c_{sx_1} + f_2^*(x_1)$ | | | $f_1^*(s)$ | $x_1^*$ |
|---|---|---|---|---|---|
| | B | C | D | | |
| A | 13 | 11 | 11 | 11 | C or D |

# Prototype Example

- Subway Transfer Problem
  - The final transfer route is
    - ✓ A-C-E-H-J
    - ✓ A-D-E-H-J
    - ✓ A-D-F-I-J

# Characteristics of DP Problems

- 1. The problem can be divided into stages, with a policy decision required at each stage.
  - DP problems require a sequence of interrelated decisions.
- 2. Each stage has a number of states associated with the beginning of that stage.
  - The states are the various possible conditions.

# Characteristics of DP Problems

- 3. The effect of the policy decisions at each stage is to transform the current state to a state associated with the beginning of the next stage.
  - DP can be interpreted in terms of the networks.
    - ✓ Each node would correspond to a state.
    - ✓ The network would consist of columns of nodes with each column corresponding to a stage.
    - ✓ The links from a node to nodes in the next column corresponds to the possible policy decisions on which state to go to next.
    - ✓ The value assigned to each link usually can be interpreted as the intermediate contribution to the objective function.

# Characteristics of DP Problems

- 4. The solution procedure is designed to find optimal policy for the overall problem (i.e., a prescription of the optimal policy decision at each stage for each of the possible states).
  - In addition to optimal solutions for the overall problem, the results show how he proceed if he gets detoured to a state that is not on an optimal route.
  - For any problem, DP provides policy prescription of what to do under every possible circumstances.

# Characteristics of DP Problems

- 5. Given the current state, an optimal policy for the remaining stages is independent of the policy decisions adopted in previous stages.
  - Therefore, the optimal decision depends on only the current state and not on how you got there. This is the **principle of optimality for DP**.
- 6. The solution procedure begins by finding the optimal policy for the last stage.

# Characteristics of DP Problems

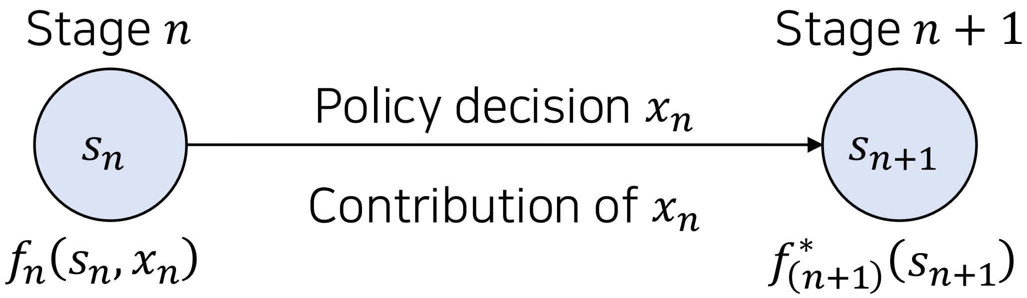
- 7. A recursive relationship that identifies the optimal policy for stage n given the optimal policy for stage n+1, is available.
  - For example, $f_n^*(s) = \min_{x_n}\{c_{sx_n} + f_{n+1}^*(x_n)\}$
    - ✓ Finding optimal policy decision which you start in states at stage n requires the minimizing value of $x_n$.
    - ✓ The corresponding minimum cost is achieved by using this value of $x_n$ and the following optimal policy when you start in state $x_n$ at stage n+1.
      - $N$: the number of stages.
      - $n$: label for current stage ($n = 1, \dots, N$)
      - $s_n$: current state for stage $n$
      - $x_n$: decision variable for stage $n$
      - $x_n^*$: optimal value of $x_n$ (given $s_n$)
      - $f_n(s_n, x_n)$: contribution of stage n, n+1, ⋯, N to objective function if system starts in state $s_n$, immediate decision is $x_n$, and optimal decisions are made thereafter.
      - $f_n^*(s_n) = f_n(s_n, x_n^*)$
      - Recursive relationship will be $f_n^*(s_n) = \max_{x_n} f_n(s_n, x_n)$

# Characteristics of DP Problems

- 8. When you use this recursive relationship, the solution procedure starts at the end and moves backward stage by stage.
  - Each time finding the optimal policy for the stage until it finds the optimal policy at the initial stage.
  - This optimal policy immediately yields an optimal solution for the entire problem.

# Deterministic DP

- Deterministic Dynamic Programming
  - The state of the next stage is determined by the current stage's state and the policy decision.




  - The form of the objective function: maximize or minimize the sum or product of contributions at each stage.

# Deterministic DP

- Deterministic Dynamic Programming
  - Example. Send 5 medical teams to 3 countries to maximize the total increase in average life expectancy.
    - ✓ Expected increase in average life expectancy by number of teams assigned to each country.

| # of medical teams | Country 1 | Country 2 | Country 3 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 45 | 20 | 50 |
| 2 | 70 | 45 | 70 |
| 3 | 90 | 75 | 80 |
| 4 | 105 | 110 | 100 |
| 5 | 120 | 150 | 130 |

    - ✓ Although the order of decisions by country is not predetermined, we can decide stage by stage.
    - ✓ Let $x_n$ be the number of teams assigned to country $n$, and $s_n$ be the number of teams remaining after assigning to previous countries.
      - $s_1 = 5, s_2 = 5 - x_1, s_3 = s_2 - x_2$

# Deterministic DP

| # medical teams | Country 1 | Country 2 | Country 3 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 45 | 20 | 50 |
| 2 | 70 | 45 | 70 |
| 3 | 90 | 75 | 80 |
| 4 | 105 | 110 | 100 |
| 5 | 120 | 150 | 130 |

- Deterministic DP
  - Example. Send 5 medical teams to 3 countries
    - ✓ Maximize Z = $\sum_{i=1}^{3} p_i(x_i)$
      - (subject to) $\sum_{i=1}^{3} x_i = 5$
      - $x_j \geq 0$ and $x_j$ is an integer.
    - ✓ $f_n(s_n, x_n) = p_n(x_n) + \max \sum_{i=n+1}^{3} p_i(x_i)$
    - ✓ $\sum_{i=n}^{3} x_i = s_n$
    - ✓ $f_n^*(s_n) = \max_{x_n \in \{0,\ldots,s_n\}} f_n(s_n, x_n)$
    - ✓ $f_n(s_n, x_n) = p_n(x_n) + f_{n+1}^*(s_n - x_n)$
    - ✓ $f_3^*(s_3) = \max_{x_3} p_3(x_3)$

# Deterministic DP

- Deterministic DP
  - Most common type: effort allocation
    - ✓ Allocate a single resource efficiently across multiple activities.
  - Example. Assigning research teams
    - ✓ Three research teams (1, 2, 3) are currently working on the same project, with failure probabilities 0.4, 0.6, and 0.8, respectively. By assigning additional scientists to each team, these failure probabilities can be reduced. The objective is to maximize the probability that at least one team succeeds.

| # of additional scientists | Research team 1 | Research team 2 | Research team 3 |
|---|---|---|---|
| 0 | 0.4 | 0.6 | 0.8 |
| 1 | 0.2 | 0.4 | 0.5 |
| 2 | 0.15 | 0.2 | 0.3 |

# Q&A