

# AI 알고리즘

## 동적계획법

# 동적계획법 적용 사례

동적계획법 적용 사례(배낭문제)

## 동적계획법 적용 사례(배낭문제)

### ❖ 배낭문제

- 배낭의 무게 10kg

| 물건 | 1  | 2  | 3  | 4  |
|----|----|----|----|----|
| 무게 | 5  | 3  | 6  | 4  |
| 가치 | 10 | 60 | 20 | 40 |

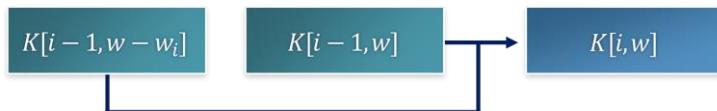
- 시간 개념이 아니어도 가능

물건 1~( $i - 1$ )까지 고려하여  
현재 배낭의 용량이  $w$ 인  
경우의 최대 가치



- 상태 정의

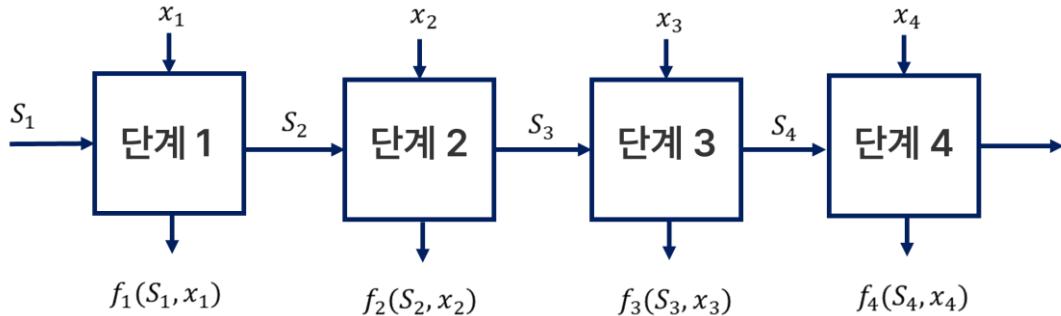
- 공간이 얼마 남아 있는가?
- 물건을 넣을 것인가?
- 수식 표현



$$K[i, w] = \begin{cases} K[i-1, w] & \text{if } w_i > w \\ \max\{v_i + K[i-1, w-w_i], K[i-1, w]\} & \text{else} \end{cases}$$

## 동적계획법 적용 사례(배낭문제)

### ❖ 배낭문제



- $f_n(i)$ : 단계 n의 상태 i에서 시작하여 최종 단계까지 최적 정책을 사용할 때 얻는 이익
- $f_n(i) = \max\{v_n x_n + f_{n+1}(i - w_n x_n)\}$

| 단계 n | 상태 $S_n$ 의 의사결정 |                          | 최적값 | 최적 정책 j | 탐색 순서 |
|------|-----------------|--------------------------|-----|---------|-------|
| 4    | 0-3             | -                        | 0   | $x_4=0$ | *     |
|      | 4-7             | 40                       | 40  | $x_4=1$ |       |
|      | 8-10            | 80                       | 80  | $x_4=2$ |       |
| 3    | 0-3             | -                        | 0   | $x_3=0$ | *     |
|      | 4-5             | -                        | -   | $x_3=0$ |       |
|      | 6-9             | 20                       | 20  | $x_3=1$ |       |
|      | 10              | 20+40                    | 60  | $x_3=1$ |       |
| 2    | 0-2             | -                        | -   | $x_2=0$ |       |
|      | 3-5             | 60                       | 60  | $x_2=1$ |       |
|      | 6-8             | 120                      | 120 | $x_2=2$ |       |
|      | 9-10            | 180                      | 180 | $x_2=3$ | *     |
| 1    | 10              | (0) 180 (1) 10+60 (2) 20 | 180 | $x_1=0$ | *     |

- 4kg밖에 없어서 0~3까지는 신경 쓸 필요 없음
- 최저 정책은 한 개 집어넣음

## 동적계획법 적용 사례(배낭문제)

### ❖ 선형계획법 문제

$$\text{MAX } 150,000x_1 + 100,000x_2$$

s.t

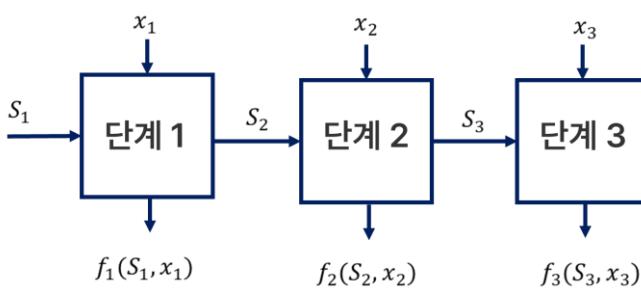
$$2x_1 + x_2 \leq 16$$

$$x_1 + x_2 \leq 12$$

$$x_1 \leq 6$$

$$x_1, x_2 \geq 0$$

- 단계(Stage) 결정
  - 각 변수를 하나의 단계로 정의
- 각 단계에서의 상태 결정
  - 각 제약식에서의 잔여 용량으로 상태 정의
    - $S_n = \{Y_1^n, Y_2^n, Y_3^n\}$
    - $Y_i^n$ : 단계 n에서 자원 i의 잔여량
- 각 단계 n의 상태 값  $S_n$ 에서의 의사결정 변수  $D_n$  결정
  - $D_n = x_n$
  - $X_n$ 이 실행가능해가 되기 위한 제약식 조건 필요
- 반복식
  - 단계 n에서 상태  $S_n$ 에서의 의사결정 변수  $D_n$ 값을 최적으로 정하기 위한 반복식으로 표현

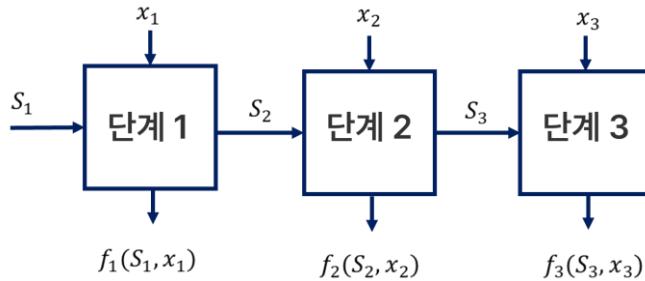


$$f_n^*(s) = \text{Max or Min}\{f_n(S_n, x_n)\}$$

$$f_2(S_2, x_2) = C_2(x_2) + f_3^*(S_3)$$

## 동적계획법 적용 사례(배낭문제)

### ❖ 선형계획법 문제



$$f_n^*(s) = \text{Max or Min} \{f_n(s_n, x_n)\}$$

$$f_2(S_2, x_2) = C_2(x_2) + f_3^*(S_3)$$

$$f_2^*(S_2) = \text{Max} \{f_3^*(S_3) + r_2(S_2, D_2)\}$$

$$= \text{Max} \{r_2(S_2, D_2)\}$$

$$\begin{aligned} & \text{Max} \\ & \begin{array}{l} x_2 \leq Y_1^2 \\ x_2 \leq Y_2^2 \\ x_2 \geq 0 \end{array} \quad \{100,000x_2\} \end{aligned}$$

- 제약 조건 다 만족해야 함
- 목적 함수식이 얼마나 올라가느냐 목적 함수에 있는 그대로 작성

$$\begin{aligned} f_1^*(S_1) &= \text{Max} \{f_2^*(S_2) + r_1(S_1, D_1)\} \\ &= \text{Max} \{f_2^*(S_2) + 150,000x_1\} \\ &\begin{aligned} & \text{Max} \\ & \begin{array}{l} 2x_1 \leq 16 \\ x_1 \leq 12 \\ x_1 \leq 6 \\ x_1 \geq 0 \end{array} \quad \{f_2^*(16 - 2x_1, 12 - x_1, 6 - x_1) + 150,000x_1\} \end{aligned} \end{aligned}$$

## 동적계획법 적용 사례(배낭문제)

### ❖ 선형계획법 문제

$$\begin{aligned}
 f_1^*(S_1) &= \max \{f_2^*(S_2) + r_1(S_1, D_1)\} \\
 &= \max \{f_2^*(S_2) + 150,000x_1\} \\
 &\quad \begin{array}{l} \max \\ 2x_1 \leq 16 \\ x_1 \leq 12 \\ x_1 \leq 6 \\ x_1 \geq 0 \end{array} \\
 &= \max \{f_2^*(16 - 2x_1, 12 - x_1, 6 - x_1) + 150,000x_1\}
 \end{aligned}$$

#### ■ n=2인 경우

- 최대화하고, 제약식을 만족하는  $x_2 = \min\{Y_1^2, Y_2^2\}$

$$\begin{array}{l} \max \\ x_2 \leq Y_1^2 \\ x_2 \leq Y_2^2 \\ x_2 \geq 0 \end{array} \\
 f_2^*(S_2) = \max \{100,000x_2\}$$

#### ■ n=1인 경우

- 제약식을 만족하는  $x_1 = 0 \leq x_1 \leq 6$

$$\begin{array}{l} \max \\ 2x_1 \leq 16 \\ x_1 \leq 12 \\ x_1 \leq 6 \\ x_1 \geq 0 \end{array} \\
 f_1^*(S_1) = \max \{f_2^*(16 - 2x_1, 12 - x_1, 6 - x_1) + 150,000x_1\}$$

$$\begin{aligned}
 f_1^*(S_1) &= \max_{0 \leq x_1 \leq 6} \{f_2^*(16 - 2x_1, 12 - x_1, 6 - x_1) + 150,000x_1\} \\
 &= \max_{0 \leq x_1 \leq 6} \{100,000 \min\{16 - 2x_1, 12 - x_1\} + 150,000x_1\} \\
 &= \begin{cases} 100,000(12 - x_1) + 150,000x_1, & 0 \leq x_1 \leq 4 \\ 100,000(16 - 2x_1) + 150,000x_1, & 4 < x_1 \leq 6 \end{cases}
 \end{aligned}$$

- $x_1 = 4$  일 때, 최댓값 1,400,000

## 동적계획법 적용 사례(배낭문제)

### ❖ 선형계획법 문제

- n=2인 경우

$$f_2^*(S_2) = \begin{array}{l} \underset{\substack{x_2 \leq Y_1^2 \\ x_2 \leq Y_2^2 \\ x_2 \geq 0}}{\text{Max}} \\ \{100,000x_2\} \end{array}$$

-  $x_2 \leq 16 - 2x_1 = 8$ ,  $\lfloor x_2 \rfloor \leq 12 - x_1 = 8$ ,  $x_2 \geq 0$ 을 만족하는  $x_2$  는  $0 \leq x_2 \leq 8$

$$f_2^*(S_2) = \begin{array}{l} \underset{0 \leq x_2 \leq 8}{\text{Max}} \\ \{100,000x_2\} \end{array}$$

- $x_2=8$  일 때, 최대
- 따라서,  $x_1=4, x_2=8$ 일 때 최댓값 1,400,000

# 동적계획법 적용 사례

와그너-위튼 모델  
(Wagner-Whitin: W-W)

## 와그너-위튼 모델(Wagner-Whitin: W-W)

### ❖ 와그너-위튼(Wagner-Whitin: W-W) 모델

- 시간에 따른 수요량이 변동되는 상황에서 최소의 비용으로 주어진 수요량을 만족시키기 위한 생산 계획을 수립하는 문제
- 이미 존재하는 도메인 → 응용하여 사용

$$\text{Min } \sum_{t=1}^T C_t(x_t)$$

s.t

$$I_t = I_{t-1} + x_t - d_t$$

$$I_t, x_t \geq 0, \forall t \in T$$

| 용어         |                                |
|------------|--------------------------------|
| $T$        | 생산계획 기간                        |
| $I_t$      | $t$ 시점 초의 재고 수준                |
| $x_t$      | $t$ 시점에서의 생산량                  |
| $d_t$      | $t$ 시점에서의 수요량                  |
| $C_t(x_t)$ | $t$ 시점에서 $x_t$ 만큼을 생산함에 따른 총비용 |

- 생산계획 기간  $T$  동안의 각 이산 시점  $t$ 에 주어진 수요량을 최소의 비용으로 100% 만족시키기 위한 각 시점  $t$  별 생산계획  $\{x_t\}$ 를 찾는 문제
  - 수요가 없으면 판매하지 못하며, 수요를 맞춰주어야 함
- 자동차 페인트 예시

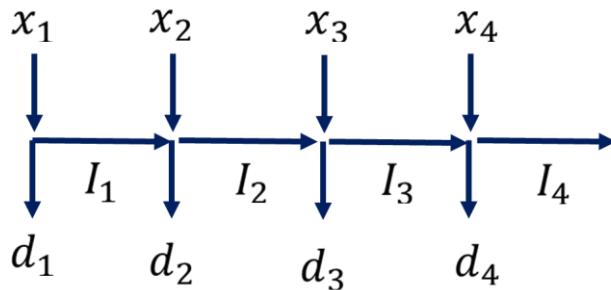


- 재고 부족비용은 발생하지 않고, 추후 납품(Backorder)을 허용하지 않는 모형

## 와그너-위튼 모델(Wagner-Whitin: W-W)

### ❖ 와그너-위튼(Wagner-Whitin: W-W) 모델

- 들어오는 값의 합과 나가는 값의 합이 같음



$$I_{t-1} + x_t = d_t + I_t$$

- 최적해가 한번 규명이 되면 문제를 쉽게 풀 수 있음
- 목적함수

$$C_t(x_t) = S_t \delta(x_t) + p_t(x_t) + h_t(I_t)$$

- $\delta(x)$ :  $x > 0$  이면 1, 아니면 0인 이진 변수
- $S_t$ : 시점 t에서의 준비 비용
- $p_t(x_t)$ :  $x_t$ 만큼을 생산할 때의 생산 비용
- $h_t(I_t)$ : 시점 t에서  $I_t$ 만큼 재고를 유지하는데 드는 재고 유지 비용

## 와그너-위튼 모델(Wagner-Whitin: W-W)

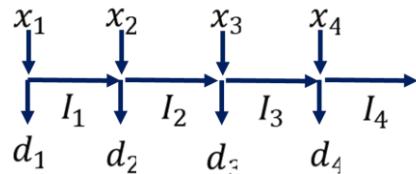
### ❖ 와그너-위튼(Wagner-Whitin: W-W) 모델

- 재고균형 방정식(Inventory Balance Equation)
  - 생산되어서 입고되는 제품양과 수요에 의해 출고되는 제품 양 간에는 항상 균형이 유지된다는 것을 나타내는 식

$$\begin{aligned}
 I_{t-1} + x_t &= d_t + I_t \\
 I_1 &= I_0 + x_1 - d_1 \\
 I_2 &= I_1 + x_2 - d_2 = (I_0 + x_1 - d_1) + x_2 - d_2 \\
 &= I_0 + (x_1 + x_2) - (d_1 + d_2) \\
 I_t &= I_{t-1} + x_t - d_t = I_{t-1} + \sum_{j=1}^t x_i - \sum_{j=1}^t d_i
 \end{aligned}$$

- 단계(Stage) 결정
  - 각 시점  $t$ 로 정의
- 각 단계에서의 상태 결정
  - 재고 수준  $I_t$ 로 상태 정의
- 각 단계  $n$ 의 상태값  $S_n$ 에서의 의사결정변수  $D_n$  결정
  - $D_n = x_t$
  - W-W 모델의 최적해는  $I_{(t-1)} = 0$  for all  $t$

$$x_t = 0 \text{ or } \sum_{j=t}^k d_j \quad \text{for } t \leq k \leq T$$



- 만약 어떤 시점  $t$ 에서의 재고수준  $I_{(t-1)} = 0$ 이면

$$I_t = I_{t-1} + x_t - d_t = x_t - d_t = 0 \text{ or } \sum_{j=t+1}^k d_j$$

## 와그너-위튼 모델(Wagner-Whitin: W-W)

### ❖ 와그너-위튼(Wagner-Whitin: W-W) 모델

- 각 단계  $n$ 의 상태값  $S_n$ 에서의 의사결정변수  $D_n$  결정
  - 최적성의 원리가 만족되는 구조를 가짐
  - W-W 모델의 총 계획기간  $T$ 인 전체 문제는 기간 1부터 기간  $t-1$ 까지의 문제와 기간  $t$ 부터 기간  $T$ 까지의 문제를 독립적으로 나눠서 풀 수 있게 됨
- 반복식
  - 단계  $n$ 에서 상태  $S_n$ 에서의 의사결정변수  $D_n$ 값을 최적으로 정하기 위한 반복식으로 표현

$$f_t^*(S_t) = \min_{1 \leq j \leq t} \{f_{j-1}^*(S_{j-1}) + r_j(S_j, D_j)\}$$

$$f_0^*(S_0) = 0$$

$$r_j(S_j, D_j) = \sum_{i=j}^t C_i(x_i), \quad 1 \leq j \leq t$$

$$x_i = \begin{cases} \sum_{l=j}^t d_l, & i = j \\ 0, & i \neq j \end{cases}$$

- 와그너-위튼 모델은 문제를 단순화한 논문

## 와그너-위튼 모델(Wagner-Whitin: W-W)

### ❖ 와그너-위튼(Wagner-Whitin: W-W) 모델

- 생산 계획 기간 T=3, 초기 재고  $I_0=0$
- 비용함수

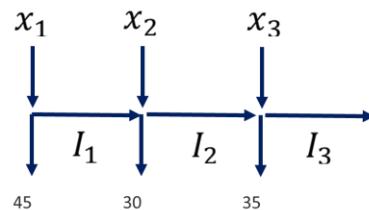
$$C_t = \begin{cases} 0, & x_t = 0 \\ a_t + h_t I_t, & x_t > 0 \end{cases}$$

- 비용요소

| 시점 t         | 1  | 2  | 3  |
|--------------|----|----|----|
| 수요량 $d_t$    | 45 | 30 | 35 |
| 준비비용 $a_t$   | 50 | 60 | 70 |
| 재고유지비용 $h_t$ | 1  | 1  | 1  |

- 생산기업 계획 기간 3 비용 함수 임의로 즘

$$\begin{aligned} \text{Min } & \sum_{t=1}^3 C_t(x_t) = C_1(x_1) + C_2(x_2) + C_3(x_3) \\ \text{s. t. } & \\ & I_1 = 0 + x_1 - d_1 \\ & I_2 = I_1 + x_2 - d_2 \\ & I_3 = I_2 + x_3 - d_3 \\ & I_t, x_t \geq 0, \quad t = 1, 2, 3 \end{aligned}$$



- 3개 더한 110 이후 생산할 필요 없음

- 반복식

$$\begin{aligned} f_t^* &= \min_{1 \leq j < t} \{f_{j-1}^* + C_j(\sum_{i=j}^t d_i), f_{t-1}^* + C_t(d_t)\} \\ f_0^* &= 0, \quad I_i = \sum_{l=i+1}^t d_l \end{aligned}$$