Optimal Design of Energy Systems (M2794.003400)

# Chapter 10. Dynamic Programming

**Min Soo KIM**

**Department of Mechanical and Aerospace Engineering**
**Seoul National University**

ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

# Chapter 10. Dynamic Programming

## 10.1 Uniqueness of Dynamic Programming Problems

- One of optimization method, applicable either to

  1. Staged processes
  2. Continuous function, approximated by staged processes.

- "Dynamic" : No connection with the frequent use of the word (e.g. "동적인")
- Related with the calculus of variations, whose result is an **optimal function**
- Finite-step of dynamic programming = Approximation of the calculus of variation

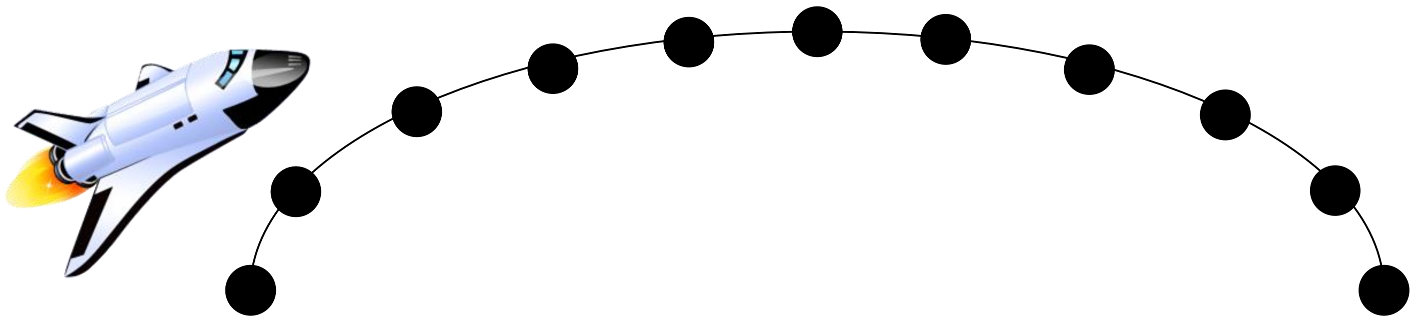
# Chapter 10. Dynamic Programming

## 10.1 Uniqueness of Dynamic Programming Problems

- For example, when determining the trajectory of a spacecraft in minimum fuel cost in terms of dynamic programming

  1. Divide the total path into **a number of segments**

  2. Then, consider **the continuous function** as a series of stages.



# Chapter 10. Dynamic Programming

## 10.2 Symbolic Description of Dynamic Programming

- The result is **optimized summation**, denoted as $\sum_{i=1}^{n} r_i$ , while the result of the calculus of variation is expressed in an integral.

S : Input to each stage

r : Return from a stage

S′ : Output from each stage

d : Decision variable

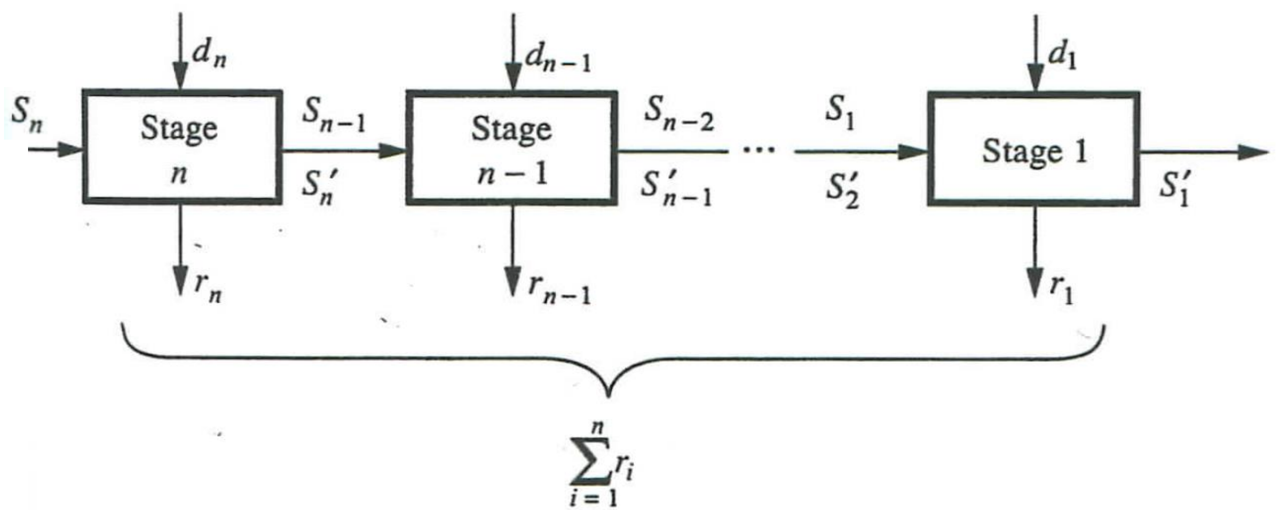



**Fig**. Pictorial representation of problem that can be solved by dynamic programming.

# Chapter 10. Dynamic Programming

## 10.3 Characteristics of The Dynamic Programming Solution

- Establishing optimal plans for subsections of the problem is the trademark of dynamic programming.
- The mechanics (or feature) is illustrated by the optimal route problem as in Example 10.1

# Chapter 10. Dynamic Programming

## Example 10.1 : Minimize the Cost using Dynamic Programming

- A pipeline is to be built between A and E, passing through one node of each B, C, and D.   Find the optimal route in the minimum total cost.

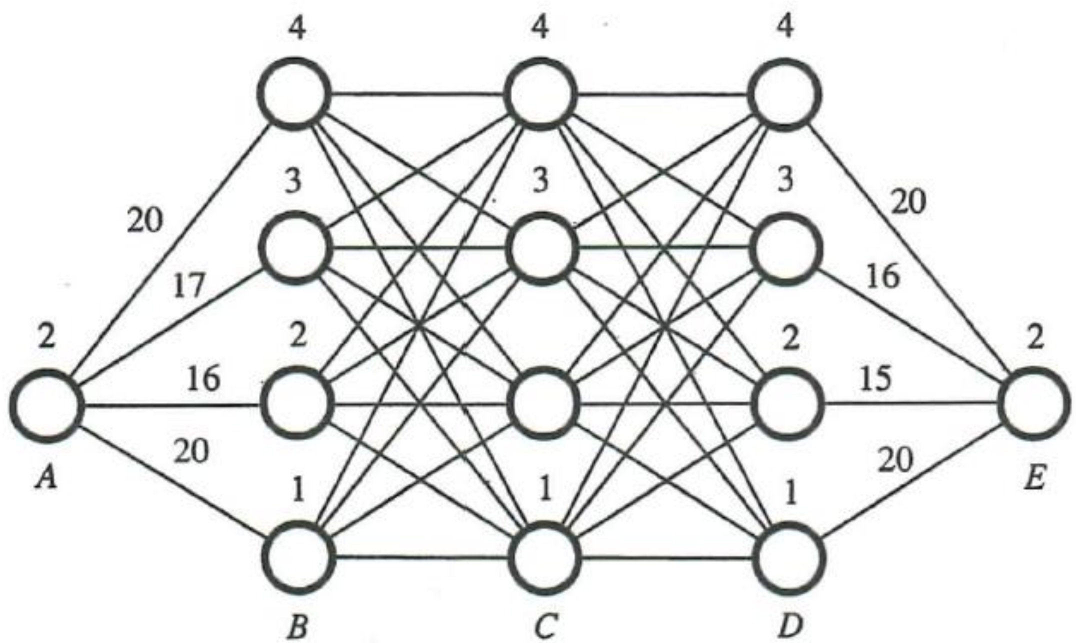



**Fig**. Dynamic programming used to minimize the cost between points A and E.

# Chapter 10. Dynamic Programming

## Example 10.1 : Minimize the Cost using Dynamic Programming

**(Given)**

- The costs of A - B and D - E are given in figure
- The costs of B – C and C - D are given in table

**Table** Costs from B to C and C to D in Fig.

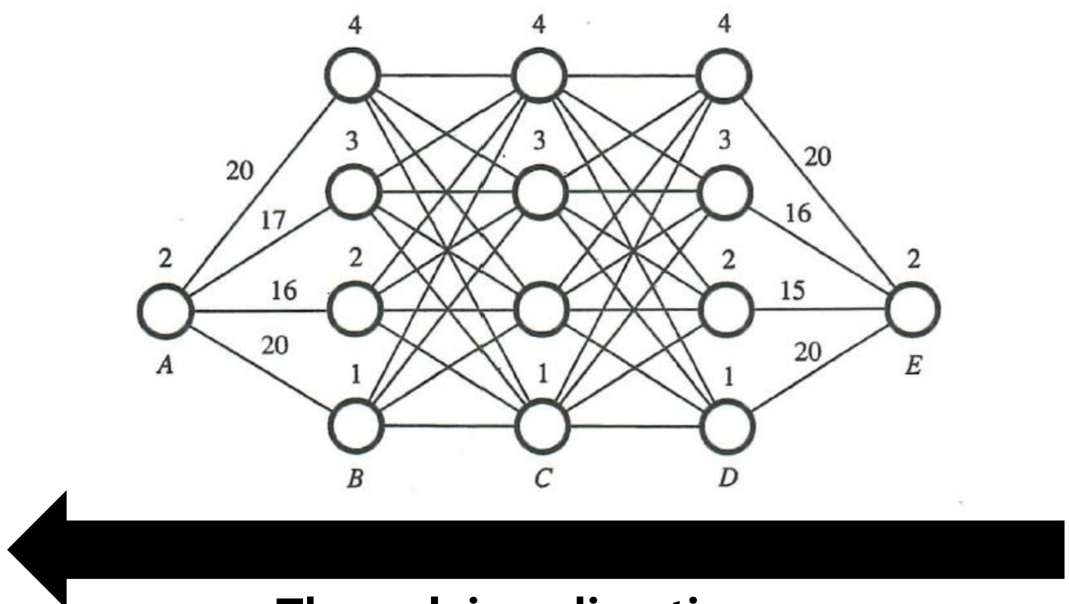
| From | To 1 | To 2 | To 3 | To 4 |
|---|---|---|---|---|
| 1 | 12 | 15 | 21 | 28 |
| 2 | 15 | 16 | 17 | 24 |
| 3 | 21 | 17 | 16 | 15 |
| 4 | 28 | 24 | 15 | 12 |

# Chapter 10. Dynamic Programming

## Example 10.1 : Minimize the Cost using Dynamic Programming

**(Solution)**

- We start at the right end to left, that is, from **point E to A**.
- So, the last table takes the totally optimized cost of the entire system.




**The solving direction**

# Chapter 10. Dynamic Programming

## Example 10.1 : Minimize the Cost using Dynamic Programming

### (Solution)

**Table** Example 10.1, C to E

| From | Through | Cost | | | |
|---|---|---|---|---|---|
| | | C to D | D to E | Total | Optimum |
| C4 | D4 | 12 | 20 | 32 | |
| | **D3** | **15** | **16** | **31** | √ |
| | D2 | 24 | 15 | 39 | |
| | D1 | 28 | 20 | 48 | |
| C3 | D4 | 15 | 20 | 35 | |
| | **D3** | **16** | **16** | **32** | √ |
| | **D2** | **17** | **15** | **32** | √ |
| | D1 | 21 | 20 | 41 | |
| C2 | D4 | 24 | 20 | 44 | |
| | D3 | 17 | 16 | 33 | |
| | **D2** | **16** | **15** | **31** | √ |
| | D1 | 15 | 20 | 35 | |
| C1 | D4 | 28 | 20 | 48 | |
| | D3 | 21 | 16 | 37 | |
| | **D2** | **15** | **15** | **30** | √ |
| | D1 | 12 | 20 | 32 | |

**Table** Example 10.1, B to E

| From | Through | Cost | | | |
|---|---|---|---|---|---|
| | | B to C | C to E | Total | Optimum |
| B4 | **C4** | **12** | **31** | **43** | √ |
| | C3 | 15 | 32 | 47 | |
| | C2 | 24 | 31 | 55 | |
| | C1 | 28 | 30 | 58 | |
| B3 | **C4** | **15** | **31** | **46** | √ |
| | C3 | 16 | 32 | 48 | |
| | C2 | 17 | 31 | 48 | |
| | C1 | 21 | 30 | 51 | |
| B2 | C4 | 24 | 31 | 55 | |
| | C3 | 17 | 32 | 49 | |
| | C2 | 16 | 31 | 47 | |
| | **C1** | **15** | **30** | **45** | √ |
| B1 | C4 | 28 | 31 | 59 | |
| | C3 | 21 | 32 | 53 | |
| | C2 | 15 | 31 | 46 | |
| | **C1** | **12** | **30** | **42** | √ |

# Chapter 10. Dynamic Programming

## Example 10.1 : Minimize the Cost using Dynamic Programming

**(Answer)**

- The optimum route : **A2 → B2 → C1 → D2 → E2**

**Table** Example 10.1, A to E

| To E from | Through | Cost | | | |
|---|---|---|---|---|---|
| | | A to B | B to E | Total | Optimum |
| A2 | B4 | 20 | 43 | 63 | |
| | B3 | 17 | 46 | 63 | |
| | **B2** | **16** | **45** | **61** | √ |
| | B1 | 20 | 42 | 62 | |

# Chapter 10. Dynamic Programming

## 10.3 Characteristics of The Dynamic Programming Solution

- **Key feature** :

  After an optimal way is determined from intermediate to final state,

  future calculations, passing through that state, use only the optimal way.

## 10.4 Efficiency of Dynamic Programming

- Dynamic programming is **efficient**, particularly in **large problems**.
- For example, consider previous an exercise problem Ex. 10.1, if one more stage is added to the problem.

Dynamic Programming : **40** routes (# of the presented in table)

Exhaustive examination : **64** routes (1×4×4×4, A-B2-C-D-E)

⬇

Dynamic Programming : **56** routes (+16, one table added )

Exhaustive examination : **256** routes (×4)

## Example 10.2 : Find the Concentrations in Minimum Cost

- A series of ultrafilters separate the protein and lactose. Use dynamic programming to solve for the concentrations leaving each stage in the minimum total cost.

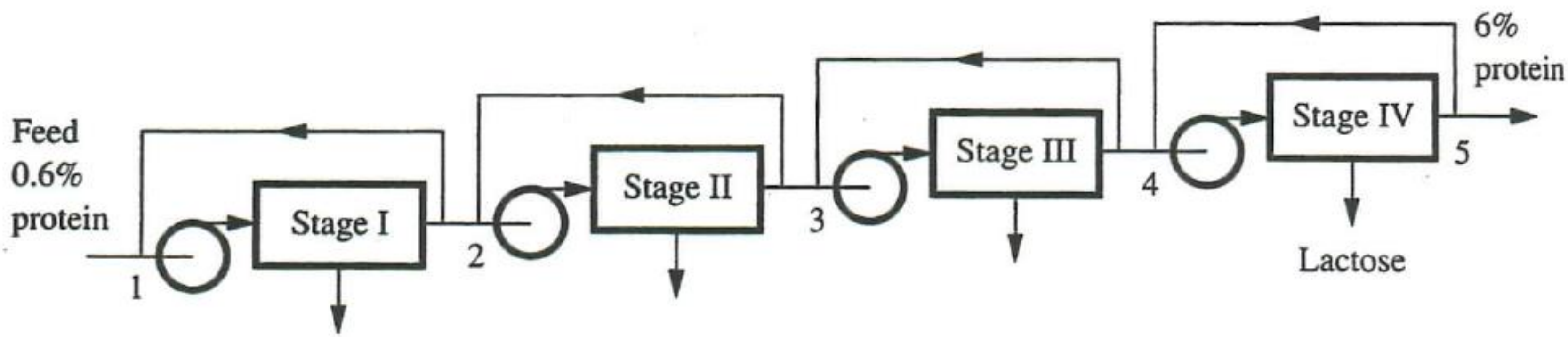



**Fig**. Chain of ultrafilters to separate protein from lactose in whey.

# Chapter 10. Dynamic Programming

## Example 10.2 : Find the Concentrations Leaving in Minimum Cost

**(Given)**

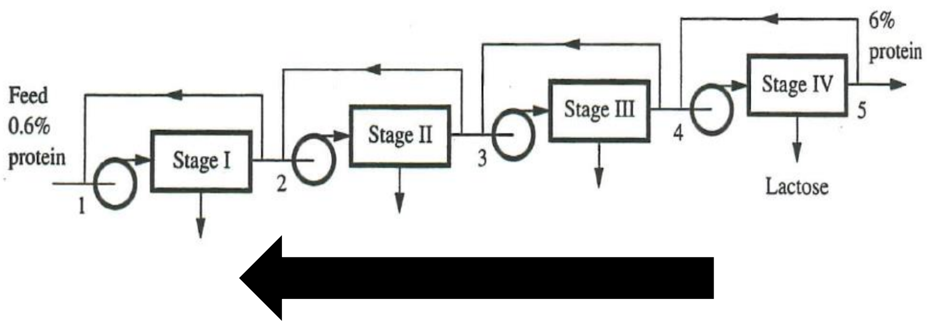
**Table** Operating cost of one stage in a protein-lactose separator, dollars

| Entering protein concen-tration | Leaving protein concentration, % | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **0.9** | **1.2** | **1.8** | **2.4** | **3.0** | **3.6** | **4.2** | **4.8** | **5.4** | **6.0** |
| **0.6** | 5.53 | 10.77 | 20.24 | 28.38 | 35.20 | 40.70 | 44.88 | 47.74 | 49.28 | 49.50 |
| **0.9** | | 3.73 | 10.77 | 17.23 | 23.10 | 28.38 | 33.07 | 37.18 | 40.70 | 43.63 |
| **1.2** | | | 5.54 | 10.78 | 15.67 | 20.24 | 24.47 | 28.38 | 31.95 | 35.20 |
| **1.8** | | | | 3.74 | 7.33 | 10.79 | 14.00 | 17.23 | 20.24 | 23.10 |
| **2.4** | | | | | 2.82 | 5.55 | 8.21 | 10.80 | 13.27 | 15.67 |
| **3.0** | | | | | | 2.26 | 4.47 | 6.63 | 8.73 | 10.81 |
| **3.6** | | | | | | | 1.89 | 3.75 | 5.56 | 7.33 |
| **4.2** | | | | | | | | 1.62 | 3.21 | 4.78 |
| **4.8** | | | | | | | | | 1.42 | 2.82 |
| **5.4** | | | | | | | | | | 1.26 |
| **6.0** | | | | | | | | | | |

## Example 10.2 : Find the Concentrations Leaving in Minimum Cost

**(Solution)**

- The calculations start at the stage IV, and proceed back until the final table.
- The minimum concentration entered in stage IV is 1.8%, because at least 0.3% of protein is added each stage. That is, **0.6 → 0.9 → 1.2 → 1.8** (%)




**The solving direction**

**Table** Example 10.2, stage IV

| Concentration Entering stage IV (%) | Through | Cost ($) |
|---|---|---|
| 1.8 | - | 23.10 |
| 2.4 | - | 15.67 |
| 3.0 | - | 10.81 |
| 3.6 | - | 7.33 |
| 4.2 | - | 4.78 |
| 4.8 | - | 2.82 |
| 5.4 | - | 1.26 |

## Example 10.2 : Find the Concentrations Leaving in Minimum Cost

(Solution)

**Table** Example 10.2, stage Ⅲ and IV

| Concentration entering Ⅲ (%) | Through | Cost ($) |
|---|---|---|
| 1.2 | 1.8 | 5.54+23.10=28.64 |
| | 2.4 | **10.78+15.67=26.45*** |
| | 3.0 | 15.67+10.81=26.48 |
| | 3.6 | 20.24+7.33=27.57 |
| | 4.2 | 24.47+4.78=29.25 |
| | 4.8 | 28.38+2.82=31.20 |
| | 5.4 | 31.95+1.26=33.21 |
| 1.8 | 2.4 | 3.74+15.67=19.41 |
| | 3.0 | 7.33+10.81=18.14 |
| | 3.6 | **10.79+7.33=18.12*** |
| | 4.2 | 14.00+4.78=18.78 |
| | 4.8 | 17.23+2.82=20.05 |
| | 5.4 | 20.24+1.26=21.50 |
| 2.4 | 3.0 | 2.82+10.81=13.63 |
| | 3.6 | **5.55+7.33=12.88*** |
| | 4.2 | 8.21+4.78=12.99 |
| | 4.8 | 10.80+2.82=13.62 |
| | 5.4 | 13.27+1.26=14.53 |
| 3.0 | 3.6 | 2.26+7.33=9.59 |
| | 4.2 | **4.47+4.78=9.25*** |
| | 4.8 | 6.63+2.82=9.45 |
| | 5.4 | 8.73+1.26=9.99 |
| 3.6 | 4.2 | 1.89+4.78=6.67 |
| | 4.8 | **3.75+2.82=6.57*** |
| | 5.4 | 5.56+1.26=6.82 |
| 4.2 | 4.8 | **1.62+2.82=4.44*** |
| | 5.4 | 3.21+1.26=4.47 |
| 4.8 | 5.4 | 1.42+1.26=2.68 |

## Example 10.2 : Find the Concentrations Leaving in Minimum Cost

### (Solution)

**Table** Example 10.2, stage Ⅱ, Ⅲ and IV

| Concentration entering Ⅱ (%) | Through | Cost ($) | Concentration entering Ⅱ (%) | Through | Cost ($) |
|---|---|---|---|---|---|
| 0.9 | 1.2 | 3.73+26.45=30.18 | 1.8 | 2.4 | 3.74+12.88=16.62 |
| | 1.8 | **10.77+18.12=28.89*** | | 3.0 | **7.33+9.25=16.58*** |
| | 2.4 | 17.23+12.88=30.11 | | 3.6 | 10.79+6.57=17.36 |
| | 3.0 | 23.10+9.25=32.35 | | 4.2 | 14.00+4.44=18.44 |
| | 3.6 | 28.38+6.57=34.95 | | 4.8 | 17.23+2.68=19.91 |
| | 4.2 | 33.07+4.44=37.51 | 2.4 | 3.0 | **2.82+9.25=12.07*** |
| | 4.8 | 37.18+2.68=39.86 | | 3.6 | 5.55+6.57=12.12 |
| 1.2 | 2.4 | **5.57+18.12=23.66*** | | 4.2 | 8.21+4.44=12.65 |
| | 3.0 | **10.78+12.88=23.66*** | | 4.8 | 10.80+2.68=13.48 |
| | 3.6 | 15.67+9.25=24.92 | 3.0 | 3.6 | **2.26+6.57=8.83*** |
| | 4.2 | 20.24+6.57=26.81 | | 4.2 | 4.47+4.44=8.91 |
| | 4.8 | 24.47+4.44=28.91 | | 4.8 | 6.63+2.68=9.31 |
| | 5.4 | 28.38+2.68=31.06 | 3.6 | 4.2 | **1.89+4.44=6.33*** |
| | | | | 4.8 | 3.75+2.68=6.43 |
| | | | 4.2 | 4.8 | **1.62+2.68=4.30*** |

# Chapter 10. Dynamic Programming

## Example 10.2 : Find the Concentrations Leaving in Minimum Cost

**(Answer)**

- The system has the minimum cost at **0.6 → 0.9 → 1.8 → 3.6 → 6 (%)**

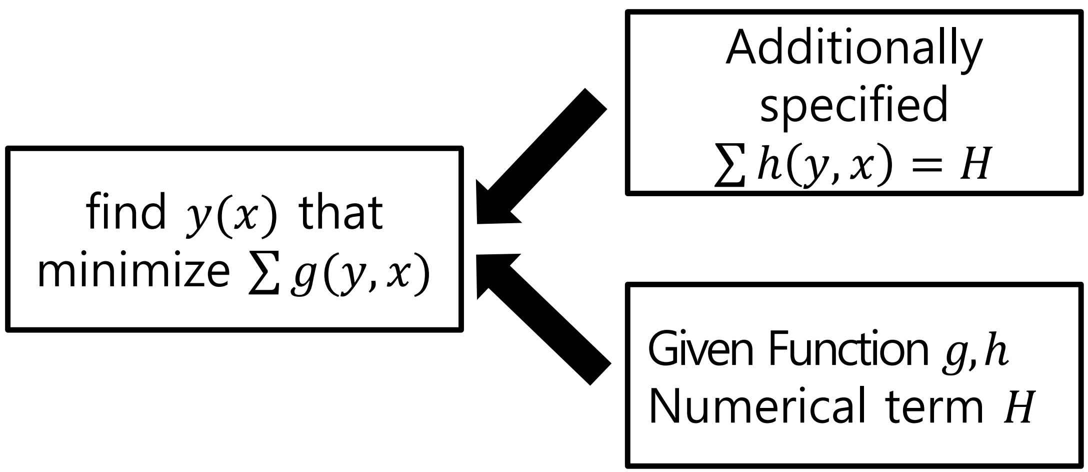
**Table** Example 10.2, stage Ⅰ and Ⅳ

| Concentration entering Ⅰ (%) | Through | Cost ($) |
|---|---|---|
| 0.6 | 0.9 | **5.53+28.89=34.42*** |
| | 1.2 | 10.77+23.66=34.43 |
| | 1.8 | 20.24+16.58=36.82 |
| | 2.4 | 28.38+12.07=40.45 |
| | 3.0 | 35.20+8.83=44.03 |
| | 3.6 | 40.70+6.33=47.03 |
| | 4.2 | 44.88+4.30=49.18 |

# Chapter 10. Dynamic Programming

## 10.6 Apparently Constrained Problems

- Constrained optimization : Optimization problem + Constrained condition
- Constrained problem can be converted to unconstrained case, that will be covered in Example 10.3.



# Chapter 10. Dynamic Programming

**Example 10.3 : Decide the distribution of tubes to minimize pressure drop**

- An evaporator which boils liquid inside tubes consists of 4 banks of tubes. Determine the distribution of the 40 tubes so that the total pressure drop in the evaporator is minimum using dynamic programming.

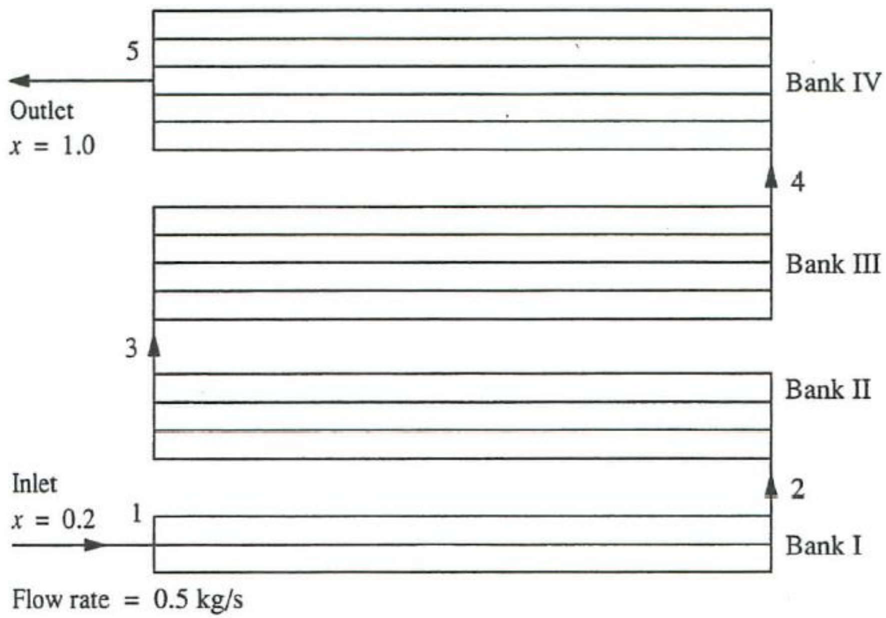



**Fig**. Evaporator in Example 10.3.

# Chapter 10. Dynamic Programming

## Example 10.3 : Decide the distribution of tubes to minimize pressure drop

**(Given)**

- The flow rate : $\dot{m}_{in} = 0.5\, kg/s$ , $\dot{m}_{vaporizing} = 0.01\, kg/s$ (each tube)
- A fraction of vapor : $x_{in} = 0.2$ , $x_{out} = 1$

  increasing $x$ by 0.02
- The pressure drop : $\Delta p = 720\, (\frac{x_i}{n})^2$ [kPa]

  $n$ : number of tubes in bank, $x_i$ : vapor fraction entering bank

# Chapter 10. Dynamic Programming

## Example 10.3 : Decide the distribution of tubes to minimize pressure drop

**(Solution)**

- Choose the state variable **cumulative(누적량) tubes** as shown in figure below.
- Before stage Ⅰ, no tubes have been committed, and following stage IV, all of tubes, 40, have been committed.
- Entering stage Ⅰ, vapor fraction : $x = 0.2$, pressure drop : $\Delta p = 720\,(\frac{x_i}{n})^2$ [kPa]

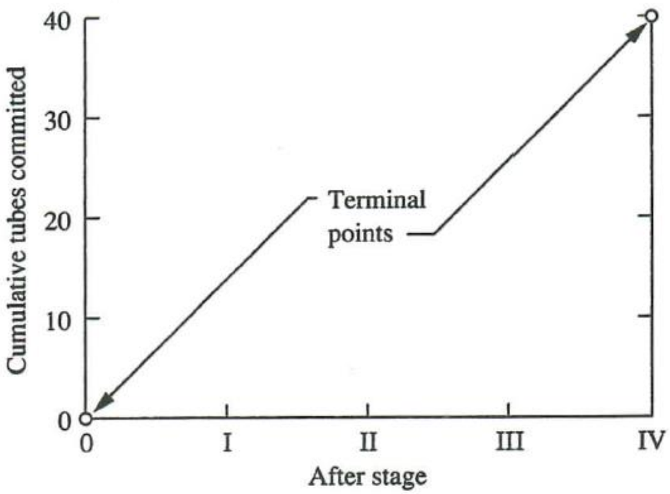



**Fig**. State variables of cumulative number of tubes committed in Example 10.3.

**Table** Example 10.3, stage Ⅰ

| Total tubes committed | Tubes in Stage Ⅰ (n) | Total $\Delta \boldsymbol{p}$ (kPa) |
|---|---|---|
| 2 | 2 | 7.20 |
| 3 | 3 | 3.20 |
| 4 | 4 | 1.80 |
| 5 | 5 | 1.15 |
| 6 | 6 | 0.80 |

# Chapter 10. Dynamic Programming

## Example 10.3 : Decide the distribution of tubes to minimize pressure drop

### (Solution)

- Entering stage Ⅱ, vapor fraction : $x_i = 0.2 + 0.02 \times (number\ of\ tubes\ in\ \text{Ⅰ})$

**Table** Example 10.3, stage Ⅰ and Ⅱ

| Total tubes committed | Tubes in StageⅡ (n) | Total $\Delta p$ (kPa) | Total tubes committed | Tubes in StageⅡ (n) | Total $\Delta p$ (kPa) |
|---|---|---|---|---|---|
| 11 | 5 | 0.80+2.95=3.75 | 13 | 7 | 0.80+1.50=2.30 |
| | 6 | **1.15+1.80=2.95*** | | 8 | **1.15+1.01=2.16*** |
| | 7 | **1.80+1.15=2.95*** | | 9 | 1.80+0.73=2.53 |
| | 8 | 3.20+0.76=3.96 | | 10 | 3.20+0.49=3.69 |
| | 9 | 7.20+0.51=7.70 | 14 | 7 | 0.59+1.70=2.29 |
| 12 | 6 | 0.80+2.05=2.85 | | 8 | **0.80+1.15=1.95*** |
| | 7 | **1.15+1.32=2.47*** | | 9 | **1.15+0.80=1.95*** |
| | 8 | 1.80+0.88=2.68 | | 10 | 1.80+0.56=2.36 |
| | 9 | 3.20+0.60=3.68 | 15 | 8 | 0.59+1.30=1.89 |
| | | | | 9 | **1.15+0.80=1.71*** |
| | | | | 10 | 1.15+0.65=1.80 |

## Example 10.3 : Decide the distribution of tubes to minimize pressure drop

### (Solution)

- Entering stage Ⅲ, vapor fraction : $x_i = 0.2 + 0.02 \times (number\ of\ tubes\ cumulated)$

**Table** Example 10.3, stage Ⅰ and Ⅲ

| Total tubes committed | Tubes in StageⅢ (n) | Total $\Delta \boldsymbol{p}$ (kPa) | Total tubes committed | Tubes in StageⅢ (n) | Total $\Delta \boldsymbol{p}$ (kPa) |
|---|---|---|---|---|---|
| 22 | 9 | 2.16+1.88=4.04 | 25 | 10 | 1.71+1.80=3.51 |
| | 10 | **2.47+1.39=3.86*** | | 11 | 1.95+1.37=3.32 |
| | 11 | 2.95+1.05=4.00 | | 12 | **2.16+1.06=3.22*** |
| 23 | 9 | 1.95+2.05=4.00 | | 13 | 2.47+0.82=3.29 |
| | 10 | 2.16+1.52=3.68 | 26 | 11 | 1.71+1.49=3.20 |
| | 11 | **2.47+1.15=3.62*** | | 12 | **1.95+1.15=3.00*** |
| | 12 | 2.95+0.88=3.83 | | 13 | 2.16+0.90=3.06 |
| 24 | 10 | 1.95+1.66=3.61 | | | |
| | 11 | **2.16+1.26=3.42*** | | | |
| | 12 | 2.49+0.97=3.44 | | | |
| | 13 | 2.95+0.75=3.70 | | | |

# Chapter 10. Dynamic Programming

## Example 10.3 : Decide the distribution of tubes to minimize pressure drop

**(Answer)**

- The optimal distribution of tubes is **5, 7, 11, 17** at stage Ⅰ, Ⅱ, Ⅲ, IV, respectively.
- The total pressure drop is 4.71 kPa

**Table** Example 10.3, stage Ⅰ to IV

| Total tubes committed | Tubes in Stage IV(n) | Total $\Delta p$ (kPa) |
|---|---|---|
| 40 | 13 | 2.93+2.33=5.26 |
| | 14 | 3.00+1.90=4.90 |
| | 15 | 3.22+1.57=4.79 |
| | 16 | 3.42+1.30=4.72 |
| | 17 | 3.62+1.09=4.71* |
| | 18 | 3.86+0.91=4.77 |

# Chapter 10. Dynamic Programming

## 10.7 Summary

- It is suitable to optimize a system that consists of a chain of components where the **output, from a unit, forms the input to next**.
- It can be more efficient when calculating in **large systems**.
- Challenge appears in setting up tables and identifying the **state variables**.