Dynamic Programming

Dynamic Programming

- **Transforms a complex optimization problem into a sequence of smaller ones (divide and conquer)**
- **Problem is divided into a sequence of stages**
- **Each stage has associated states**
- **Relies on recursion and principle of optimality**
- **Forward or backward pass to optimize**
- **Most general of all solution methodologies**

Example: Resource Allocation Problem

A company has \$6000 to invest and 3 investments are available. If d_j thousands of dollars are invested in investment j, then a net present value (in thousands) of $r_j(d_j)$ is obtained.

In particular,

$$r_1(d_1) = \begin{cases} 7d_1 + 2 & \mathrm{d}_1 > 0 \\ 0 & \mathrm{d}_1 = 0 \end{cases}$$

$$r_2(d_2) = \begin{cases} 3d_2 + 7 & \mathrm{d}_2 > 0 \\ 0 & \mathrm{d}_2 = 0 \end{cases}$$

$$r_3(d_3) = \begin{cases} 4d_3 + 5 & \mathrm{d}_3 > 0 \\ 0 & \mathrm{d}_3 = 0 \end{cases}$$

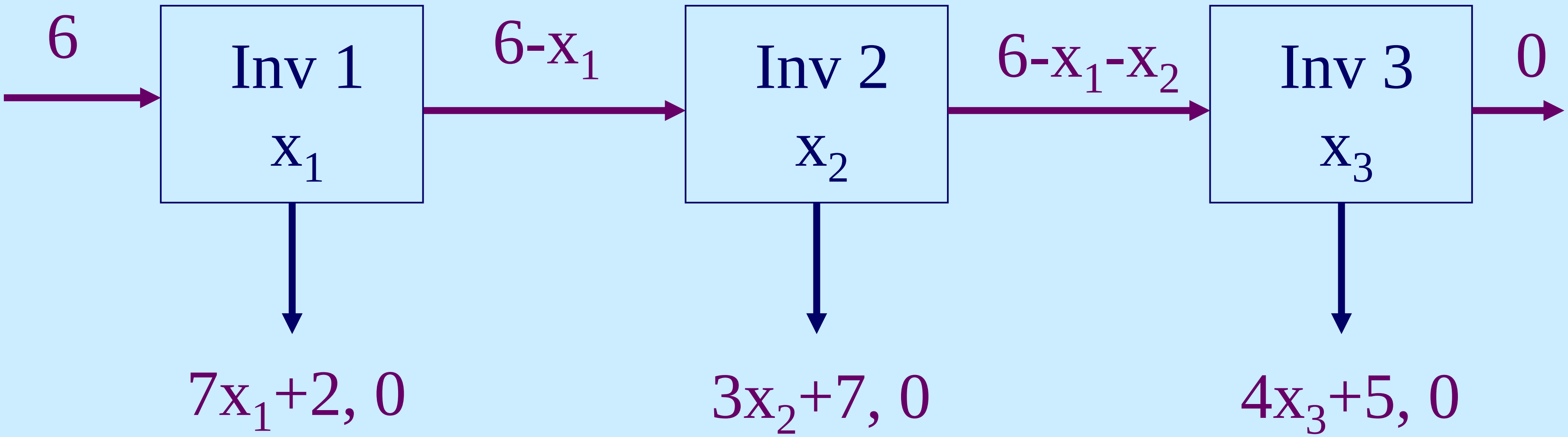
Example Cont.

The amount placed in each investment must be an integer multiple of \$1000.

? How should the company allocate \$6000 to maximize the net present value

$$\begin{aligned} \max \quad & r_1(x_1) + r_2(x_2) + r_3(x_3) \\ s.t. \quad & x_1 + x_2 + x_3 = 6 \\ & x_1, x_2, x_3 \geq 0 \quad \text{and integer} \end{aligned}$$

DP Functions



In general, n such investments:

$f_k(d)$ = max net present value by investing d dollars in investments k, k+1, k+2, …, n

$f_1(6)$: optimal value

compute using f_3, f_2, and f_1 values

DP Recursion, Principle of Optimality, and Boundary Conditions

$$f_k(d) = \underset{\substack{0 \le x_k \le d: \\ x_k \text{ integer}}}{\text{maximize}} \{r_k(x_k) + f_{k+1}(d - x_k)\} \quad \text{for k} \le \text{n-1}$$

Principle of optimality: optimal decision at stage k should use optimal decision of later stages

$$f_n(d) = r_n(d) \qquad \text{for d} \ge 0$$

To trace the best solution:

Let $x_k(d)$ = amount to invest in alternative k if d dollars are available for alternatives k, k+1, …, n

Solving with DP

1) Identify DP Function

$f_k(d)$ = max net present value by investing d dollars in investments k, k+1, k+2, …, n

2) Construct Recursion

$$f_k(d) = \underset{\substack{0 \le x_k \le d: \\ x_k \text{ integer}}}{\text{maximize}} \{r_k(x_k) + f_{k+1}(d - x_k)\} \quad \text{for } k \le \text{n-1}$$

3) Keep track of best solution

$x_k(d)$ = amount to invest in alternative k if d dollars are available for alternatives k, k+1, …, n

Solving with DP

4) Identify the base case (initiate recursion)

$$f_n(d) = r_n(d) \qquad \text{for } d \geq 0$$

5) Identify optimal value

$f_1(n)$: optimal value

6) Starting with the base case solve for all necessary function values

Solutions to Recursive Functions (Stage 3)

$$f_3(0) = 0 \qquad x_3(0) = 0$$

$$f_3(1) = 9 \qquad x_3(1) = 1$$

$$f_3(2) = 13 \qquad x_3(2) = 2$$

$$f_3(3) = 17 \qquad x_3(3) = 3$$

$$f_3(4) = 21 \qquad x_3(4) = 4$$

$$f_3(5) = 25 \qquad x_3(5) = 5$$

$$f_3(6) = 29 \qquad x_3(6) = 6$$

Solutions to Recursive Functions (Stage 2)

$$f_2(d) = \text{maximize} \quad \{r_2(x_2) + f_3(d - x_2)\}$$

$$0 \le x_2 \le d:$$

$$x_2 \ \text{integer}$$

$$\mathrm{x}_2(d) = \text{best value above}$$

$$f_2(0) = 0 \qquad x_2(0) = 0$$

$$f_2(1) = 10 \qquad x_2(1) = 1$$

$$f_2(2) = 19 \qquad x_2(2) = 1$$

$$f_2(3) = 23 \qquad x_2(3) = 1$$

$$f_2(4) = 27 \qquad x_2(4) = 1$$

$$f_2(5) = 31 \qquad x_2(5) = 1$$

$$f_2(6) = 35 \qquad x_2(6) = 1$$

Solutions to Recursive Functions (Stage 1)

$$f_1(d) = \max\text{imize} \quad \{r_1(x_1) + f_2(d - x_1)\}$$
$$0 \le x_1 \le d:$$
$$x_1 \ \text{integer}$$
$$\mathrm{x}_1(d) = \text{best value above}$$

$$f_1(6) = \max\{35, 40, 43, 46, 49, 47, 44\} = 49$$
$$x_1(6) = 4$$

Trace back the solution: $x_1 = 4, \quad x_2 = 1, \quad x_3 = 1$

$(\mathrm{f}_2(2)) \quad (\mathrm{f}_3(1))$

Optimal Solution:Invest \$4000 in 1, \$1000 in 2, and \$1000 in 3

Net Present Value = \$49000

Binary Knapsack Problem

$$\max \quad \sum_{j=1}^{n} c_j x_j$$

$$s.t. \quad \sum_{j=1}^{n} a_j x_j \quad \le \; b$$

$$x_j \; \in \; \{0,1\} \text{ for j=1, ..., n}$$

DP Function:

$$f_k(d) = \max \quad \sum_{j=1}^{k} c_j x_j \qquad \text{for d=0,1,...b}$$

$$s.t. \quad \sum_{j=1}^{k} a_j x_j \quad \le \; d \qquad \text{k=1,2,...n}$$

$$x_j \; \in \; \{0,1\} \text{ for j=1, ..., k}$$

Optimal Value: $f_n(b)$

Binary Knapsack Problem

Boundary Conditions:
$$f_1(d) = \begin{cases} c_1 & \text{if} \quad a_1 \le d \\ 0 & \text{otherwise} \end{cases}$$

Recursion using principle of optimality: *for* k=2,...n

b=0,...,b

$$f_k(d) = \begin{cases} f_{k\text{-}1}(d) & \text{if} \quad a_k > d \\ \max\{\underbrace{f_{k\text{-}1}(d)}_{x_k=0} \;,\; \underbrace{c_k + f_{k\text{-}1}(d - a_k)}_{x_k=1}\} & \text{if} \quad a_k \le d \end{cases}$$

To trace: Let $x_k(d)$ be the best value of this variable

Example: Binary Knapsack Problem

$$\max \quad 16x_1 + 19x_2 + 23x_3 + 28x_4$$

$$s.t. \quad 2x_1 + 3x_2 + 4x_3 + 5x_4 \leq 7$$

$$x_i \in \{0,1\} \quad \text{for} \quad i=1,\dots,4$$

$$f_1(d) = \begin{cases} 0 & \text{if} \quad d=0,1 & x_1 = 0 \\ 16 & \text{if} \quad d \geq 2 & x_1 = 1 \end{cases}$$

$$f_2(d) = \begin{cases} 0 & \text{if} \quad d=0,1 & x_2 = 0 \\ 16 & \text{if} \quad d = 2 & x_2 = 0 \\ 19 & \text{if} \quad d = 3,4 & x_2 = 1 \\ 35 & \text{if} \quad d = 5,6,7 & x_2 = 1 \end{cases}$$

Example: Binary Knapsack Problem

$$\begin{array}{llr} \max & 16x_1 + 19x_2 + 23x_3 + 28x_4 & \\ s.t. & 2x_1 + 3x_2 + 4x_3 + 5x_4 & \leq 7 \\ & x_i \in \{0,1\} \quad \text{for} \quad i=1,\dots,4 & \end{array}$$

$$f_3(d) = \begin{cases} f_2(\mathrm{d}) & \text{if} \quad \mathrm{d}=0,1,2,3 \qquad \mathrm{x}_3 = 0 \\ 23 & \text{if} \quad \mathrm{d}=4 \qquad \mathrm{x}_3 = 1 \\ 35 & \text{if} \quad \mathrm{d}=5 \qquad \mathrm{x}_3 = 1 \\ 39 & \text{if} \quad \mathrm{d}=6 \qquad \mathrm{x}_3 = 1 \\ 42 & \text{if} \quad \mathrm{d}=7 \qquad \mathrm{x}_3 = 1 \end{cases}$$

$$f_4(7) = \max(42, 28 + f_3(2)) = 44 \qquad \mathrm{x}_4 = 1$$

Hence, $\boxed{x_4^* = 1, \quad x_3^* = 0, \quad x_2^* = 0, \quad x_1^* = 1}$

General Knapsack Problem

$$\max \quad \sum_{j=1}^{n} c_j x_j$$

$$s.t. \quad \sum_{j=1}^{n} a_j x_j \quad \leq \; \text{b}$$

$$x_j \;\geq\; 0 \quad \text{and integer for j=1, ..., n}$$

DP Function:

$$g(w) = \max \quad \sum_{j=1}^{n} c_j x_j$$

$$s.t. \quad \sum_{j=1}^{n} a_j x_j \quad \leq \; \text{w}$$

$$x_j \;\geq\; 0 \quad \text{and integer for j=1, ..., n}$$

Optimal Value: g(b)

General Knapsack Problem

Boundary Conditions:

$$g(w) = 0 \qquad \text{for} \qquad 0 \le w < \min_i \{a_i\}$$

Recursion using principle of optimality:

$$g(w) = \max_{j:\, a_j \le w} \{c_j + g\,(w\text{-}a_j)\} \qquad \text{for } w \ge \min_j \{a_j\}$$

To trace: Let x (w) be the index of the variable chosen

Example: General Knapsack Problem

$$\begin{aligned} \max \quad & 11x_1 + 7x_2 + 12x_3 \\ s.t. \quad & 4x_1 + 3x_2 + 5x_3 \le 10 \\ & x_i \in \{0,1\} \quad \text{for} \quad i=1,\dots,3 \end{aligned}$$

$g(0) = g(1) = g(2) = 0$

$g(3) = 7$ $\quad x(3) = 2$

$g(4) = 11$ $\quad x(4) = 1$

$g(5) = 12$ $\quad x(5) = 3$

$g(6) = 14$ $\quad x(6) = 2$

$g(7) = 18$ $\quad x(7) = 1$ or 2

$g(8) = 22$ $\quad x(8) = 1$

$g(9) = 23$ $\quad x(9) = 1$ or 3

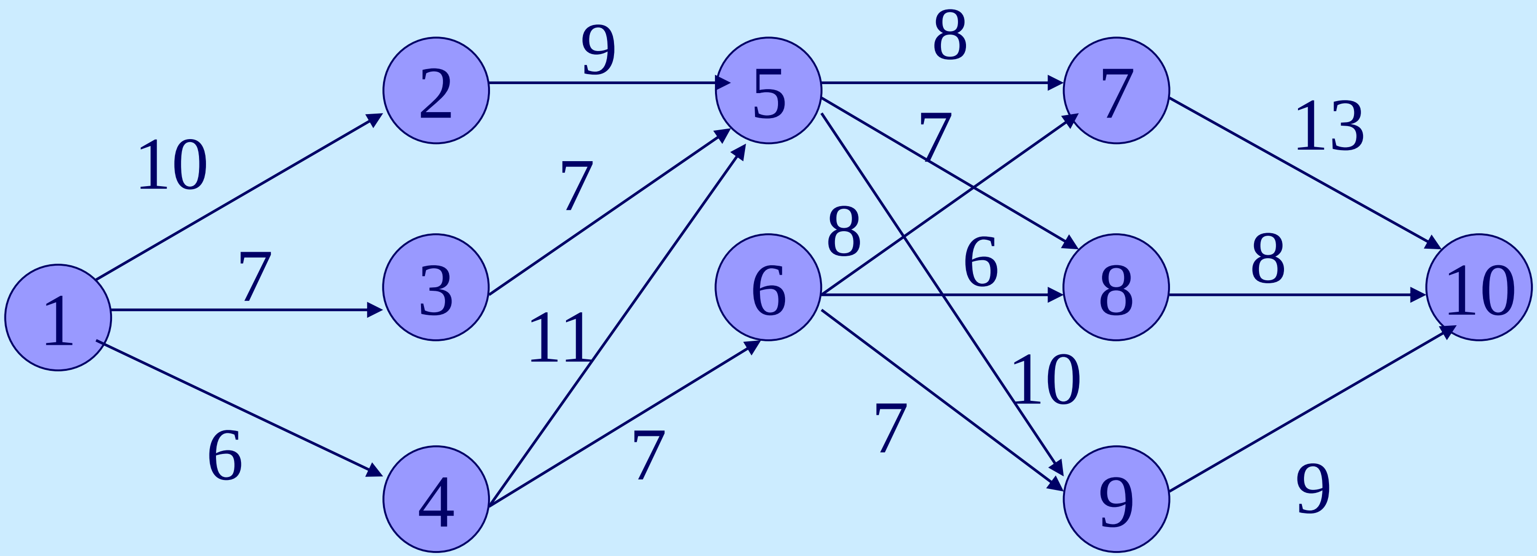
optimal value $g(10) = 25$,

x(10)=1 or 2

optimal solution:

$$x_1^* = 1, \quad x_2^* = 2, \quad x_3^* = 0$$

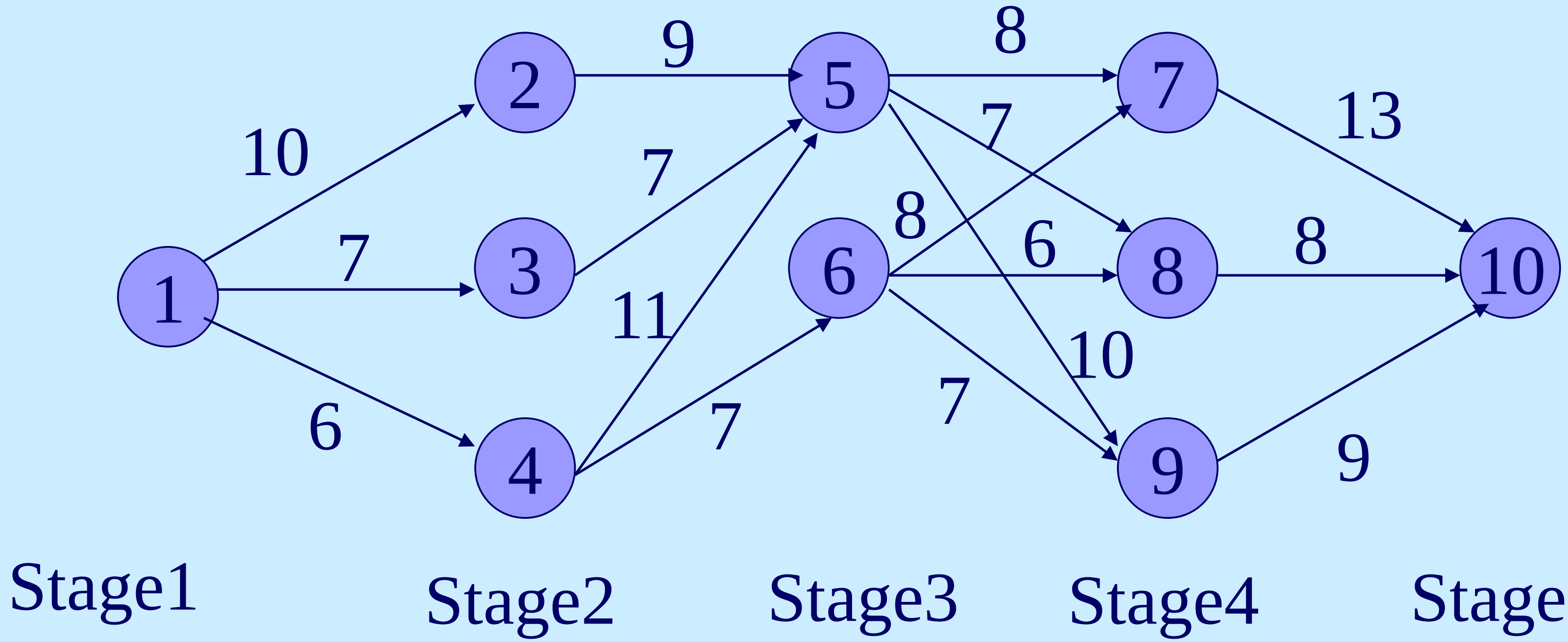
Network Example



Numbers on arcs (c_{ij}'s) correspond to altitudes G=(N,A)

?Find a path from node 1 to node 10 such that the highest altitude on this path is the minimum among the highest altitudes of all paths

Network Example



DP Function:

$f_t(i)$=the highest altitude of a path with the minimum highest altitude from city i in stage t to city 10

Optimal Value: $f_1(1)$

Network Example

Boundary Conditions:

$$f_4(7) = 13$$
$$f_4(8) = 8$$
$$f_4(9) = 9$$

Recursion using principle of optimality:

$$f_t(\text{i}) = \min_{\text{j: (i,j)}\in \text{A}} \{\max(c_{ij}\ ,\ f_{t+1}(\text{j}))\} \quad \text{t=1,2,3}$$

To trace: Let d (i) be the node chosen

Network Example

$f_3(5)=8$ d(5)=8

$f_3(6)=8$ d(6)=8

$f_2(2)=9$ d(2)=5

$f_2(3)=8$ d(3)=5

$f_2(4)=8$ d(4)=6

optimal value:

$f_1(1)=8$ d(1)=3 or 4

optimal solution:

$1 \to 3 \to 5 \to 8 \to 10$

highest altitude: 8

or

$1 \to 4 \to 6 \to 8 \to 10$

highest altitude: 8

Same Example: Network Not Acyclic

Given G=(N,A), not necessarily acyclic,
specified source node s, and destination node t,
and each arc (i,j) with altitude c_{ij}

? Solve the minimum highest altitude problem from s to t

DP Function:

$f_k^i \equiv$ best path's highest altitude encountered in going from node i to node t among all paths that use at most k arcs

Optimal Value:

f_{n-1}^s where |N|=n

Same Example: Network Not Acyclic

Boundary Conditions:

$$f_1^{\,i} = \begin{cases} c_{it} & \text{if} \quad (i,t) \in A \\ \infty & \text{otherwise} \end{cases} \qquad \text{i} \neq \text{t}$$

$$f_{k+1}^{\,i} = \min_{j:(i,j)\in A} \{\max(c_{ij}, f_k^{\,j})\} \qquad \text{k} = 1, \ldots, \text{n - 2}$$

To trace: Let d (i) be the node chosen