Deterministic Dynamic Programming

Fatih Cavdur

fatihcavdur@uludag.edu.tr

Introduction

Dynamic Programming (DP) is a technique that can be used to solve many optimization problems. In most applications, DP obtains solutions by working backward from the end of a problem toward the beginning, thus breaking up a large, unwieldy problem into a series of smaller, more tractable problems.

DP Terminology:

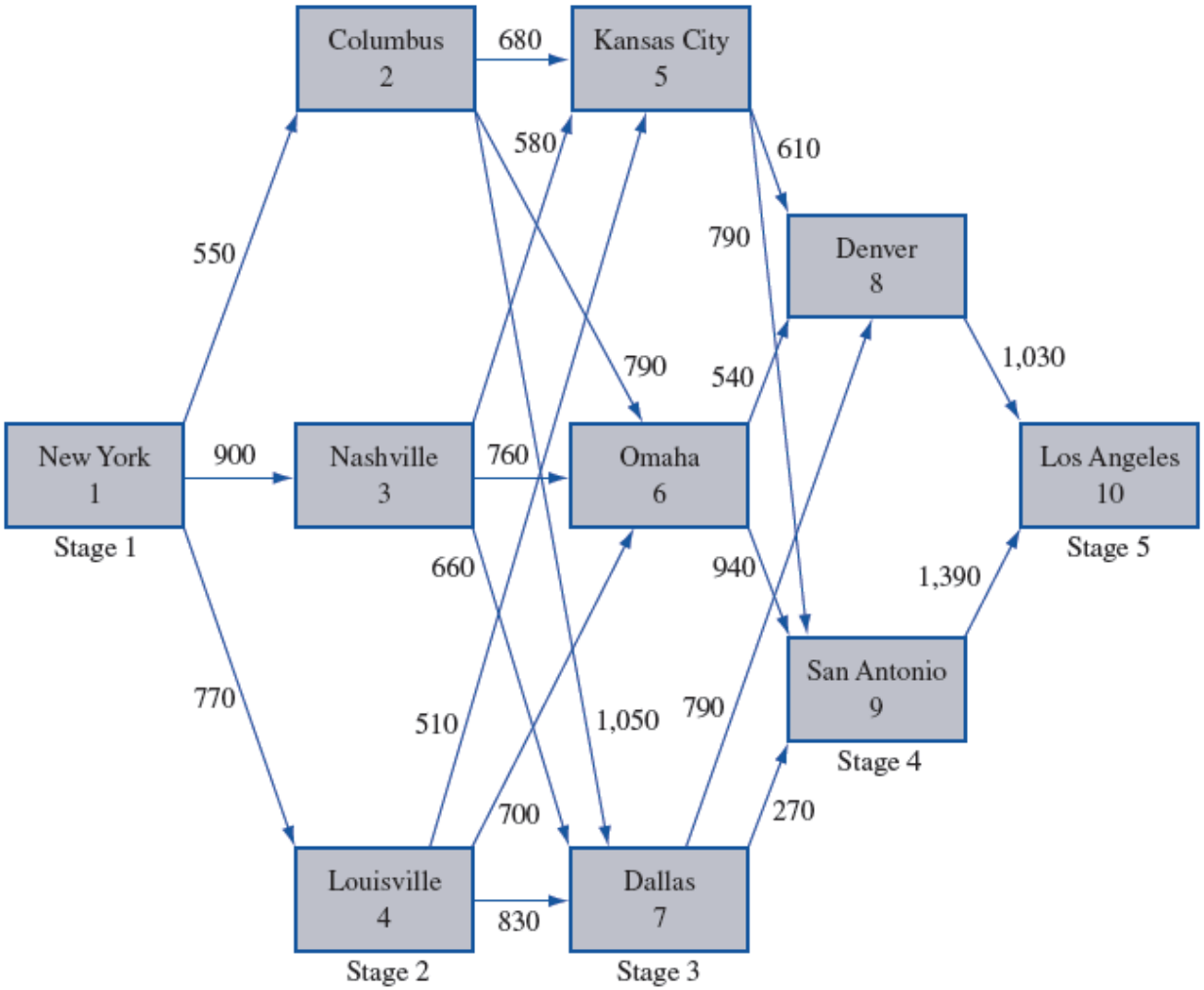
Stage	: t
State	: s_t
Decision Variables	: x_t
Optimal Decision or Policy	: $x_t^*(s_t)$
State Transformation Function	: $t_t\left(s_t, x_t^*(s_t)\right)$
Optimal Value or Objective Function	: $f_t^*(s_t)$
Immediate Contribution Function	: $c_t(s_t, x_t)$

fatihcavdur@uludag.edu.tr

Example: A Shortest Path Problem

Joe Cougar lives in New York City, but he plans to drive to Los Angeles to seek fame and fortune. Joe's funds are limited, so he has decided to spend each night on his trip at a friend's house. Joe has friends in Columbus, Nashville, Louisville, Kansas City, Omaha, Dallas, San Antonio, and Denver. Joe knows that after one day's drive he can reach Columbus, Nashville, or Louisville. After two days of driving, he can reach Kansas City, Omaha, or Dallas. After three days of driving, he can reach San Antonio or Denver. Finally, after four days of driving, he can reach Los Angeles. To minimize the number of miles traveled, where should Joe spend each night of the trip? The actual road mileages between cities are given in the below figure.

Example: A Shortest Path Problem



Example: A Shortest Path Problem

If we let,

c_{ij} = the road mileages between city i and j

$f_t(i)$ = the length of the shortest path from city i to Los Angeles, given that city i is a stage t city

Stage 4 Computations:

$$f_4(8) = 1{,}030$$

$$f_4(9) = 1{,}390$$

fatihcavdur@uludag.edu.tr

Example: A Shortest Path Problem

Stage 3 Computations:

$$f_3(5) = \min \begin{Bmatrix} c_{58} + f_4(8) = 610 + 1{,}030 = 1{,}640 \\ c_{59} + f_4(9) = 790 + 1{,}390 = 2{,}180 \end{Bmatrix} \Rightarrow x_3(5) = 8$$

$$f_3(6) = \min \begin{Bmatrix} c_{68} + f_4(8) = 540 + 1{,}030 = 1{,}570 \\ c_{69} + f_4(9) = 940 + 1{,}390 = 2{,}330 \end{Bmatrix} \Rightarrow x_3(6) = 8$$

$$f_3(7) = \min \begin{Bmatrix} c_{78} + f_4(8) = 790 + 1{,}030 = 1{,}820 \\ c_{79} + f_4(9) = 270 + 1{,}390 = 1{,}660 \end{Bmatrix} \Rightarrow x_3(7) = 9$$

Example: A Shortest Path Problem

Stage 2 Computations:

$$f_2(2) = \min \begin{Bmatrix} c_{25} + f_3(5) = 680 + 1{,}640 = 2{,}320 \\ c_{26} + f_3(6) = 790 + 1{,}570 = 2{,}360 \\ c_{27} + f_3(7) = 1{,}050 + 1{,}660 = 2{,}710 \end{Bmatrix} \Rightarrow x_2(2) = 5$$

$$f_2(3) = \min \begin{Bmatrix} c_{35} + f_3(5) = 580 + 1{,}640 = 2{,}220 \\ c_{36} + f_3(6) = 760 + 1{,}570 = 2{,}330 \\ c_{37} + f_3(7) = 660 + 1{,}660 = 2{,}320 \end{Bmatrix} \Rightarrow x_2(3) = 5$$

$$f_2(4) = \min \begin{Bmatrix} c_{45} + f_3(5) = 510 + 1{,}640 = 2{,}150 \\ c_{46} + f_3(6) = 700 + 1{,}570 = 2{,}270 \\ c_{47} + f_3(7) = 830 + 1{,}660 = 2{,}490 \end{Bmatrix} \Rightarrow x_2(4) = 5$$

Example: A Shortest Path Problem

Stage 1 Computations:

$$f_1(1) = \min \begin{Bmatrix} c_{12} + f_2(2) = 550 + 2{,}320 = 2{,}870 \\ c_{13} + f_2(3) = 900 + 2{,}220 = 3{,}120 \\ c_{14} + f_2(4) = 770 + 2{,}150 = 2{,}920 \end{Bmatrix} \Rightarrow x_1(1) = 2$$

Optimal Path

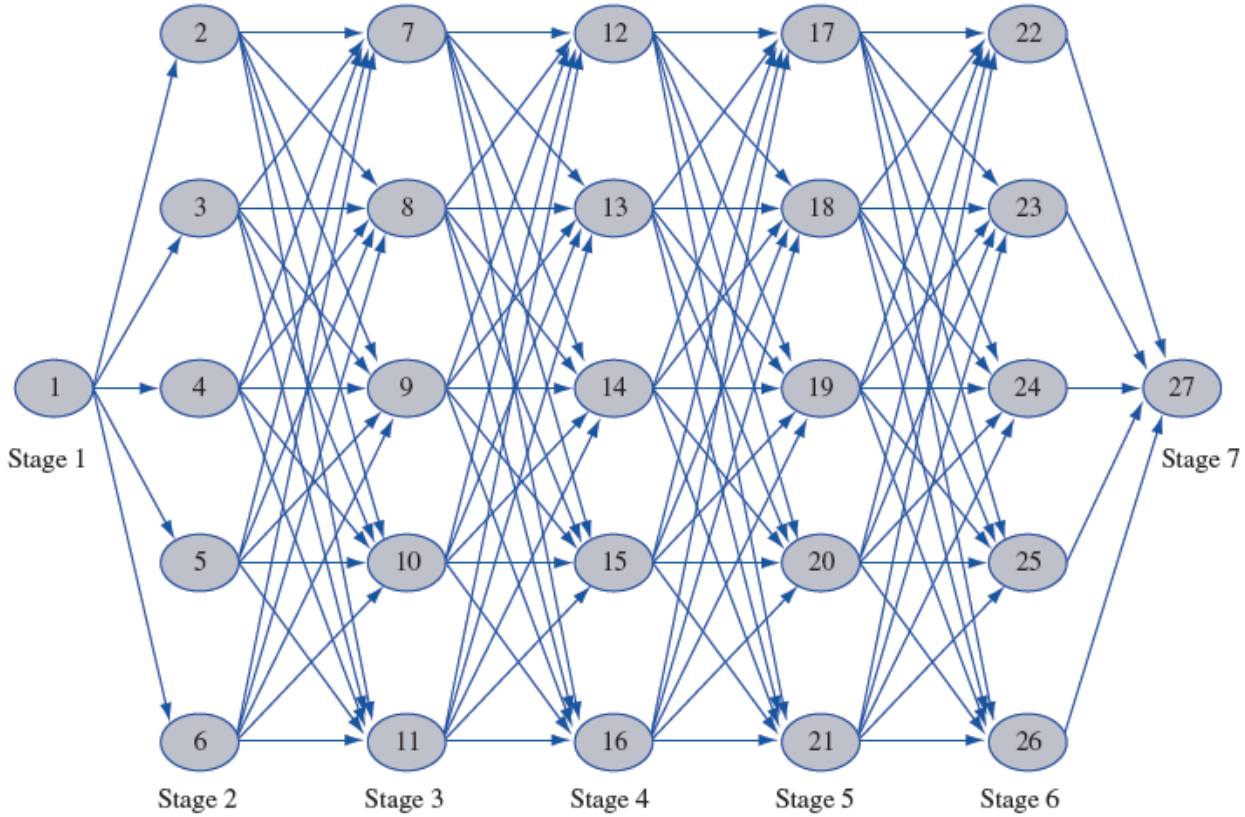
$$x_1(1) = 2; x_2(2) = 5; x_3(5) = 8; x_4(8) = 10 \Rightarrow 1 - 2 - 5 - 8 - 10;$$

fatihcavdur@uludag.edu.tr

Computational Efficiency

For the example, it would have been an easy matter to determine the shortest path from New York to Los Angeles by enumerating all the possible paths (there are only (3)(3)(2) = 18 paths). Thus, in this problem, the use of dynamic programming did not really serve much purpose. For larger networks, however, dynamic programming is much more efficient for determining a shortest path than the explicit enumeration of all paths. To see this, consider the network in Figure 2. In this network, it is possible to travel from any node in stage k to any node in stage $k+1$. Let the distance between node i and node j be c_{ij} .

Computational Efficiency



fatihcavdur@uludag.edu.tr

Computational Efficiency

Suppose we want to determine the shortest path from node 1 to node 27. If you solve this problem by explicit enumeration of all paths, there are 5^5 possible paths from node 1 to node 27. It takes five additions to determine the length of each path. Thus, explicitly enumerating the length of all paths requires $5^5 \times 5 = 5^6 = 16{,}625$ additions. When we use DP for the above example,

- Computation of $f_5(\cdot)$, $f_4(\cdot)$, $f_3(\cdot)$ and $f_2(\cdot)$ requires $5 \times 5 = 25$ additions.
- Computation of $f_1(\cdot)$ requires 5 additions.
- Thus, DP requires 4 x 25 + 5 = 105 additions to find the shortest path from node 1 to node 27

Characteristics of DP

Characteristic 1: The problem can be divided into stages with a decision required at each stage.

Characteristic 2: Each stage has a number of states associated with it. By a state, we mean the information that is needed at any stage to make an optimal decision.

Characteristic 3: The decision chosen at any stage describes how the state at the current stage is transformed into the state at the next stage.

Characteristics of DP

Characteristic 4: Given the current state, the optimal decision for each of the remaining stages must not depend on previously reached states or previously chosen decisions. This idea is known as the principle of optimality.

Characteristic 5: If the states for the problem have been classified into one of T stages, there must be a recursion that relates the cost or reward earned during stages t, $t + 1$, …, T to the cost or reward earned from stages $t + 1$, $t + 2$, …, T. In essence, the recursion formalizes the working-backward procedure.

Example DP Formulations

The owner of a lake must decide how many fishes to catch and sell each year. If she sells x fishes during year t, then a revenue $r(x)$ is earned. The cost of catching x fishes during a year is a function $c(x, b)$ of the number of fishes caught during the year and of b, the number of fishes in the lake at the beginning of the year. Of course, fishes do reproduce. To model this, we assume that the number of fishes in the lake at the beginning of a year is 20% more than the number of fishes left in the lake at the end of the previous year. Assume that there are 10,000 fishes in the lake at the beginning of the first year. Develop a dynamic programming recursion that can be used to maximize the owner's net profits over a T-year horizon.

Example DP Formulations

x_t = the number of fishes caught during year t

b_t = the number of fishes in the lake at the beginning of year t

$f_t(b_t)$ = the maximum net profit that can be earned from fishes caught during years $t, t+1, \ldots, T$ given that the number of fishes in the lake at the beginning of year t is b_t

$$f_T(b_T) = \max_{x_T}\{r(x_T) - c(x_T, b_T)\}; \quad 0 \le x_T \le b_T$$

$$f_t(b_t) = \max\{r(x_t) - c(x_t, b_t) + f_{t+1}[1.2(b_t - x_t)]\}; \quad 0 \le x_t \le b_t$$

fatihcavdur@uludag.edu.tr

Example DP Formulations

A company knows that the demand for its product during each of the next four months will be as follows: month 1, 1 unit; month 2, 3 units; month 3, 2 units; month 4, 4 units. At the beginning of each month, the company must determine how many units should be produced during the current month. During a month in which any units are produced, a setup cost of $3 is incurred. In addition, there is a variable cost of $1 for every unit produced. At the end of each month, a holding cost of 50¢ per unit on hand is incurred. Capacity limitations allow a maximum of 5 units to be produced during each month. The size of the company's warehouse restricts the ending inventory for each month to 4 units at most. The company wants to determine a production schedule that will meet all demands on time and will minimize the sum of production and holding costs during the four months. Assume that 0 units are on hand at the beginning of the first month.

Example DP Formulations

$f_t(i) =$ the minimum cost of meeting demands for months $t, t+1, \ldots, 4$ if i units are on hand at the beginning of month t

$c(x) =$ the cost of producing x units during a month

$x_t(i) =$ the production level during month t that minimizes the total cost during months $t, t+1, \ldots, 4$ if i units are on hand at the beginning of month t

fatihcavdur@uludag.edu.tr

Example DP Formulations

Stage 4 (Month 4) Computations:

$$f_4(i) = c(4-i); \quad i = 0,1,\dots,4$$

We have,

$$f_4(0) = c(4-0) = c(4) = 3+4 = 7 \Rightarrow x_4(0) = 4-0 = 4$$

$$f_4(1) = c(4-1) = c(3) = 3+3 = 6 \Rightarrow x_4(1) = 4-1 = 3$$

$$f_4(2) = c(4-2) = c(2) = 3+2 = 5 \Rightarrow x_4(2) = 4-2 = 2$$

$$f_4(3) = c(4-3) = c(1) = 3+1 = 4 \Rightarrow x_4(3) = 4-3 = 1$$

$$f_4(4) = c(4-0) = c(0) = 0+0 = 0 \Rightarrow x_4(4) = 4-4 = 0$$

fatihcavdur@uludag.edu.tr

Example DP Formulations

We can summarize the results in the following table.

i	x	$f_4(i)$	$x_4(i)$
0	4	7	4
1	3	6	3
2	2	5	2
3	1	4	1
4	0	0	0

Example DP Formulations

Stage 3 (Month 3) Computations:

$$f_3(i) = \min_x \left\{ \frac{i+x-2}{2} + c(x) + f_4(i+x-2) \right\}; \quad \begin{matrix} i = 0, \dots, 4 \\ x \in \{0, \dots, 5\} \\ 0 \le i + x - 2 \le 4 \end{matrix}$$

$$f_3(0) = \min_{x \in \{2,3,4,5\}} \left\{ \frac{0+x-2}{2} + c(x) + f_4(0+x-2) \right\}$$

$$= \min_{x \in \{2,3,4,5\}} \begin{Bmatrix} 0 + 5 + 7 = 12 \\ \frac{1}{2} + 6 + 6 = \frac{25}{2} \\ 1 + 7 + 5 = 13 \\ \frac{3}{2} + 8 + 4 = \frac{27}{2} \end{Bmatrix} = 12 \Rightarrow x_3(0) = 2$$

Computations continue similarly.

Example DP Formulations

i	x	$i+x-2$	$\frac{i+x-2}{2}+c(x)+f_4(i+x-2)$	$f_3(i)$	$x_3(i)$
0	2	0	12.0	12.0	2
0	3	1	12.5		
0	4	2	13.0		
0	5	3	13.5		
1	1	0	11.0		
1	2	1	11.5		
1	3	2	12.0		
1	4	3	12.5		
1	5	4	10.0	10.0	5
2	0	0	7.0	7.0	0
2	1	1	10.5		
2	2	2	11.0		
2	3	3	11.5		
2	4	4	9.0		
3	0	1	6.5	6.5	0
3	1	2	10.0		
3	2	3	10.5		
3	3	4	8.0		
4	0	2	6.0	6.0	0
4	1	3	9.5		
4	2	4	7.0		

Example DP Formulations

Stage 2 (Month 2) Computations:

$$f_2(i) = \min_x \left\{ \frac{i+x-3}{2} + c(x) + f_3(i+x-3) \right\}; \quad \begin{matrix} i = 0,1,\dots,4 \\ x \in \{0,\dots,5\} \\ 0 \le i+x-3 \le 4 \end{matrix}$$

$$f_2(0) = \min_{x \in \{3,4,5\}} \left\{ \frac{0+x-3}{2} + c(x) + f_3(i+x-3) \right\}$$

$$= \min_{x \in \{3,4,5\}} \begin{Bmatrix} 0+6+12 = 18 \\ \frac{1}{2}+7+10 = \frac{35}{2} \\ 1+8+7 = 16 \end{Bmatrix} = 16 \Rightarrow x_2(0) = 5$$

Computations continue similarly.

fatihcavdur@uludag.edu.tr

Example DP Formulations

i	x	$i+x-3$	$\frac{i+x-3}{2}+c(x)+f_3(i+x-3)$	$f_2(i)$	$x_2(i)$
0	3	0	18.0		
0	4	1	17.5		
0	5	2	16.0	16.0	5
1	2	0	17.0		
1	3	1	16.5		
1	4	2	15.0	15.0	4
1	5	3	16.0		
2	1	0	16.0		
2	2	0	15.5		
2	3	2	14.0	14.0	3
2	4	3	15.0		
2	5	4	16.0		
3	0	0	12.0	12.0	0
2	1	0	14.5		
3	2	2	13.0		
3	3	3	14.0		
3	4	4	15.0		
4	0	1	10.5	10.5	0
4	1	2	12.0		
4	2	3	13.0		
4	3	4	14.0		

fatihcavdur@uludag.edu.tr

Example DP Formulations

Stage 1 (Month 1) Computations:

$$f_1(i) = \min_x \left\{ \frac{i+x-1}{2} + c(x) + f_2(i+x-1) \right\}; \quad \begin{matrix} i = 0, \dots, 4; \\ x \in \{0, \dots, 5\} \\ 0 \le i + x - 1 \le 4 \end{matrix}$$

$$f_1(0) = \min_{x \in \{1,2,3,4,5\}} \left\{ \frac{0+x-1}{2} + c(x) + f_2(i+x-1) \right\}$$

$$= \min_{x \in \{1,2,3,4,5\}} \begin{Bmatrix} 0 + 4 + 16 = 20 \\ \frac{1}{2} + 5 + 15 = \frac{41}{2} \\ 1 + 6 + 14 = 21 \\ \frac{3}{2} + 7 + 12 = \frac{41}{2} \\ 2 + 8 + \frac{21}{2} = \frac{41}{2} \end{Bmatrix} = 20 \Rightarrow x_1(0) = 1$$

Computations continue similarly.

fatihcavdur@uludag.edu.tr

Example DP Formulations

i	x	$i+x-1$	$\frac{i+x-1}{2}+c(x)+f_2(i+x-1)$	$f_1(i)$	$x_1(i)$
0	1	0	20.0	20.0	1
0	2	1	20.5		
0	3	2	21.0		
0	4	3	20.5		
0	5	4	20.5		
1	0	0	16.0	16.0	0
1	1	1	19.5		
1	2	2	20.0		
1	3	3	19.5		
1	4	4	19.5		
2	0	1	15.5	15.5	0
2	1	2	19.0		
2	2	3	18.5		
2	3	4	18.5		
3	0	2	15.0	15.0	0
3	1	3	17.5		
3	2	4	17.5		
4	0	3	13.5	13.5	0
4	1	4	16.5		

fatihcavdur@uludag.edu.tr

Example DP Formulations

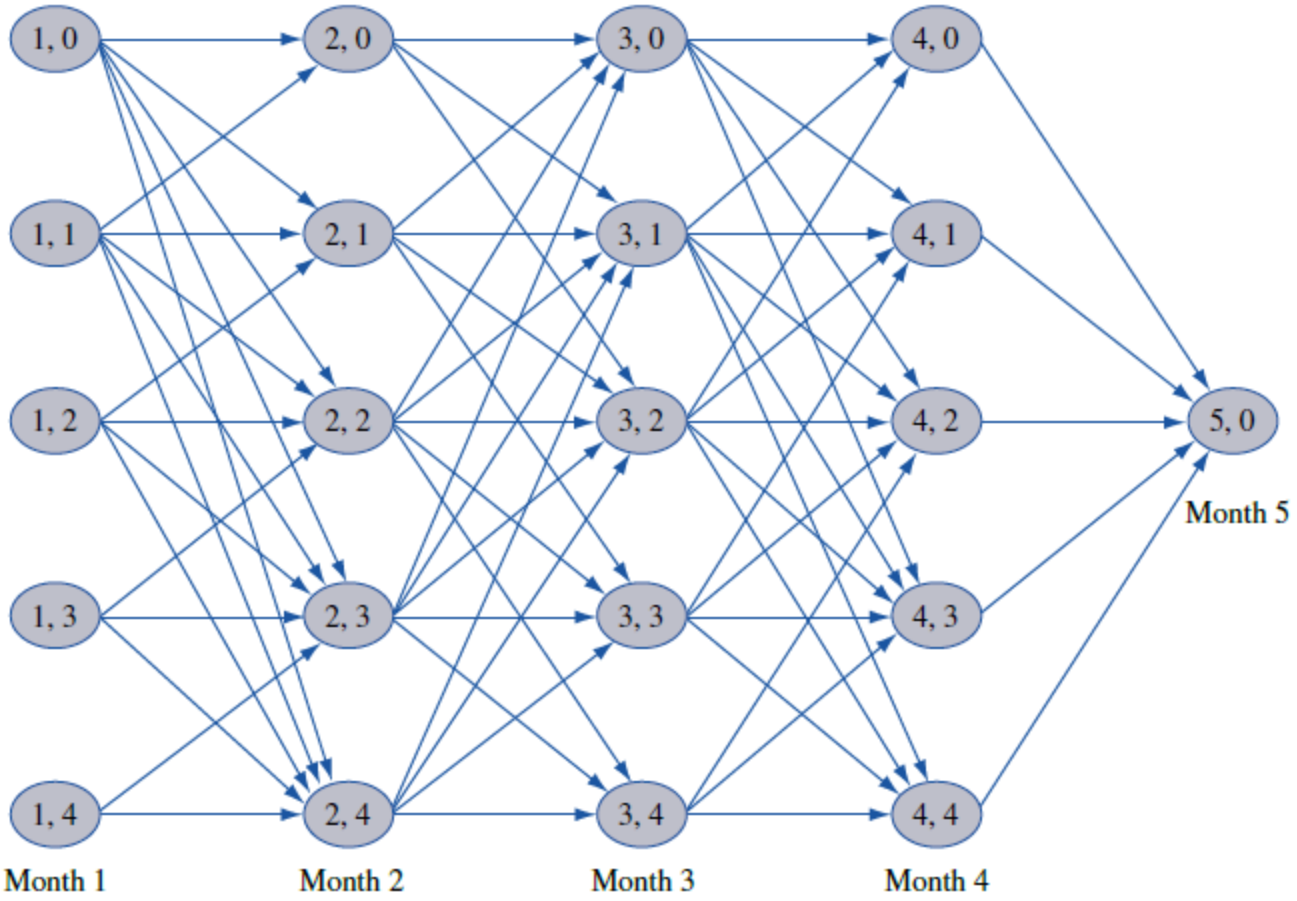
Determination of the Optimal Production Schedule

Since the initial inventory is 0 units, the minimum cost for the 4-month period will be

$$f_1(0) = 20 \Rightarrow x_1(0) = 1$$
$$f_2(0) = 16 \Rightarrow x_2(0) = 5$$
$$f_3(2) = 7 \Rightarrow x_3(2) = 0$$
$$f_4(0) = 7 \Rightarrow x_4(0) = 4$$

Thus, we should produce 1 unit during month 1, 5 units during month 2, 0 units during month 3 and 4 units during month 4 with a total cost of 20.

Example DP Formulations



Resource Allocation Problems

Assume that we have w units of available resource and T activities to which the resource can be allocated. If activity t is implemented at level x_t (non-negative integer), then, $g_t(x_t)$ units of resource are used by the activity, and a benefit $r_t(x_t)$ is obtained. The problem of determining the allocation of resources that maximizes the total benefit subject to the limited resource availability may be written as

$$\max z = \sum_{t=1}^{T} r_t(x_t)$$

$$\sum_{t=1}^{T} g_t(x_t) \leq w$$

$$x_t \in \{0,1,2,\dots\}$$

fatihcavdur@uludag.edu.tr

Resource Allocation Problems

To solve the above problem by DP, we let

$f_t(d) =$ the maximum benefit that can be obtained from activities $t, t+1, \dots, T$ if d units of the resource are available for activities $t, t+1, \dots, T$.

Thus, we can write

$$f_{T+1}(d) = 0; \quad \forall d$$

$$f_t(d) = \max\{r_t(x_t) + f_{t+1}[d - g_t(x_t)]\}; \quad x_t \in \{0,1,2, \dots\}; \quad g_t(x_t) \leq d$$

fatihcavdur@uludag.edu.tr

The Knapsack Problem

Suppose a 10-lb knapsack is to be filled with the items listed in the Table below. To maximize total benefit, how should the knapsack be filled?

Item	Weight	Benefit
1	4	11
2	3	7
3	5	12

Table: Problem Data

fatihcavdur@uludag.edu.tr

The Knapsack Problem

We let

$f_t(d) =$ the maximum benefit that can be earned from a d −pound knapsack that is filled with items of type $t, t+1, \ldots, 3$.

We have,

$$r_1(x_1) = 11x_1;\ r_2(x_2) = 7x_2;\ r_3(x_3) = 12x_3$$

$$g_1(x_1) = 4x_1;\ g_2(x_2) = 3x_2;\ g_3(x_3) = 5x_3$$

fatihcavdur@uludag.edu.tr

The Knapsack Problem

Stage 3 Computations:

$$f_3(d) = \max_{x_3}\{12x_3\}; \quad x_t \in \{0,1,2,\dots\}; \quad 5x_3 \le d$$

$$f_3(10) = 24 \Rightarrow x_3(10) = 2$$

$$f_3(5) = f_3(6) = \cdots = f_3(9) = 12 \Rightarrow x_3(5) = x_3(6) = \cdots = x_3(9) = 1$$

$$f_3(0) = f_3(1) = \cdots = f_3(4) = 0 \Rightarrow x_3(0) = x_3(1) = \cdots = x_3(4) = 0$$

fatihcavdur@uludag.edu.tr

The Knapsack Problem

$$f_2(d) = \max_{x_2}\{7x_2 + f_3(d - 3x_2)\}; \quad x_t \in \{0,1,2,\dots\}; \quad 3x_2 \le d$$

$$f_2(10) = \max\begin{Bmatrix} 7\times 0 + f_3(10) = 24 \\ 7\times 1 + f_3(7) = 19 \\ 7\times 2 + f_3(4) = 14 \\ 7\times 3 + f_3(1) = 21 \end{Bmatrix} = 24 \Rightarrow x_2(10) = 0$$

fatihcavdur@uludag.edu.tr

The Knapsack Problem

$$f_2(9) = \max \begin{Bmatrix} 7 \times 0 + f_3(9) = 12 \\ 7 \times 1 + f_3(6) = 19 \\ 7 \times 2 + f_3(3) = 14 \\ 7 \times 3 + f_3(3) = 21 \end{Bmatrix} = 21 \Rightarrow x_2(9) = 3$$

$$f_2(8) = \max \begin{Bmatrix} 7 \times 0 + f_3(8) = 12 \\ 7 \times 1 + f_3(5) = 19 \\ 7 \times 2 + f_3(2) = 14 \end{Bmatrix} = 19 \Rightarrow x_2(8) = 1$$

$$f_2(7) = \max \begin{Bmatrix} 7 \times 0 + f_3(7) = 12 \\ 7 \times 1 + f_3(4) = 7 \\ 7 \times 2 + f_3(1) = 14 \end{Bmatrix} = 14 \Rightarrow x_2(7) = 2$$

fatihcavdur@uludag.edu.tr

The Knapsack Problem

$$f_2(6) = \max\begin{Bmatrix} 7\times 0 + f_3(6) = 12 \\ 7\times 1 + f_3(3) = 7 \\ 7\times 2 + f_3(0) = 14 \end{Bmatrix} = 14 \Rightarrow x_2(6) = 2$$

$$f_2(5) = \max\begin{Bmatrix} 7\times 0 + f_3(5) = 12 \\ 7\times 1 + f_3(2) = 7 \end{Bmatrix} = 12 \Rightarrow x_2(5) = 0$$

$$f_2(4) = \max\begin{Bmatrix} 7\times 0 + f_3(4) = 0 \\ 7\times 1 + f_3(1) = 7 \end{Bmatrix} = 7 \Rightarrow x_2(4) = 1$$

$$f_2(3) = \max\begin{Bmatrix} 7\times 0 + f_3(3) = 0 \\ 7\times 1 + f_3(0) = 7 \end{Bmatrix} = 7 \Rightarrow x_2(3) = 1$$

$$f_2(2) = 7\times 0 + f_3(2) = 0 \Rightarrow x_2(2) = 0$$

$$f_2(1) = 7\times 0 + f_3(1) = 0 \Rightarrow x_2(1) = 0$$

$$f_2(0) = 7\times 0 + f_3(0) = 0 \Rightarrow x_2(0) = 0$$

fatihcavdur@uludag.edu.tr

The Knapsack Problem

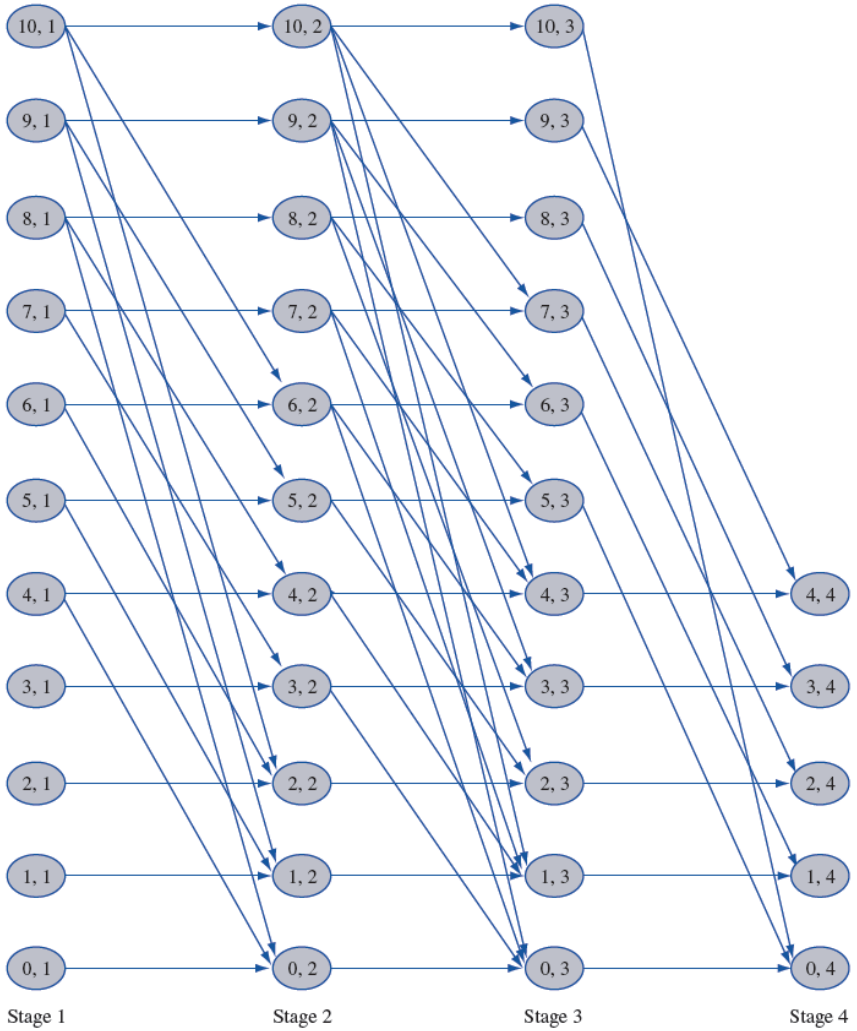
Stage 1 Computations:

$$f_1(10) = \max \begin{Bmatrix} 11 \times 0 + f_2(10) = 24 \\ 11 \times 1 + f_2(6) = 25 \\ 11 \times 2 + f_2(2) = 22 \end{Bmatrix} = 25 \Rightarrow x_1(10) = 1$$

Thus, the optimal solution is including 1 type 1 item, 2 type 2 items and 0 type 3 items as seen below:

$$f_1(10) = 10 \Rightarrow x_1(10) = 1$$
$$f_2(6) = 14 \Rightarrow x_2(6) = 2$$
$$f_3(0) = 0 \Rightarrow x_3(0) = 0$$

The Knapsack Problem



fatihcavdur@uludag.edu.tr

The Knapsack Problem

Another solution approach for the knapsack problem can be defined as follows:

- $g(w) =$ the max benefit obtained from a w-lb knapsack
- $b_j =$ benefit of item j
- $w_j =$ weight of item j

We can write, for $w = 0$, $g(0) = 0$, and for $w > 0$,

$$g(w) = \max_j \{b_j + g(w - w_j)\}$$

The Knapsack Problem

We can write, for $w = 0$, $g(0) = 0$, and for $w > 0$,

$$g(w) = \max_j \{b_j + g(w - w_j)\}$$

We can then write

$$g(0) = g(1) = g(2) = 0 \Rightarrow x(0) = x(1) = x(2) = 0$$

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

$$g(4) = \max \begin{Bmatrix} 11 + g(0) = 11 \\ 7 + g(1) = 7 \end{Bmatrix} = 11 \Rightarrow x(4) = 1$$

$$g(5) = \max \begin{Bmatrix} 11 + g(1) = 11 \\ 7 + g(2) = 7 \\ 12 + g(0) = 12 \end{Bmatrix} = 12 \Rightarrow x(5) = 3$$

The Knapsack Problem

$$g(5) = \max\begin{Bmatrix} 11 + g(1) = 11 \\ 7 + g(2) = 7 \\ 12 + g(0) = 12 \end{Bmatrix} = 12 \Rightarrow x(5) = 3$$

$$g(6) = \max\begin{Bmatrix} 11 + g(2) = 11 \\ 7 + g(3) = 14 \\ 12 + g(1) = 12 \end{Bmatrix} = 14 \Rightarrow x(5) = 2$$

If we continue similarly,

$$g(10) = \max\begin{Bmatrix} 11 + g(6) = 25 \\ 7 + g(7) = 25 \\ 12 + g(5) = 24 \end{Bmatrix} = 25 \Rightarrow x(10) = 1 \vee 2$$

Hence, one of the optimal solutions is to fill the knapsack as Type 1-Type 2-Type 2 since $x(10) = 1, x(5) = 2, x(3) = 2$.

fatihcavdur@uludag.edu.tr

The Knapsack Problem

In a knapsack problem, if we let b_i and w_i as the benefit and weight of item i, respectively, we can show that, at least one type i item will be used if $w \geq w^*$ where

$$w^* = \frac{b_i w_i}{b_i - w_i \left(\frac{b_j}{w_j}\right)}$$

and

$$\frac{b_i}{w_i} > \frac{b_j}{w_j}$$

fatihcavdur@uludag.edu.tr

The Knapsack Problem

For example, consider the following knapsack problem.

$$\max z = 16x_1 + 22x_2 + 12x_3 + 8x_4$$

$$5x_1 + 7x_2 + 5x_3 + 4x_4 \leq w$$

$$x_i \in Z^+ \quad \forall i$$

We can say that at least one Type 1 item will be in the knapsack if

$$w \geq \frac{(16)(5)}{16 - (5)\left(\frac{22}{7}\right)} = 280$$

This result, referred as a turnpike theorem, can greatly reduce the necessary computations to solve a knapsack problem.

fatihcavdur@uludag.edu.tr

The Travelling Salesman Problem

It's the last weekend of the 2004 election campaign, and candidate Walter Glenn is in New York City. Before Election Day, Walter must visit Miami, Dallas, and Chicago and then return to his New York City headquarters. Walter wants to minimize the total distance he must travel. In what order should he visit the cities? The distances in miles between the four cities are given in the Table below.

from/to	NY	Miami	Dallas	Chicago
NY	-	1,334	1,559	809
Miami		-	1,343	1,397
Dallas			-	921
Chicago				-

Table: Problem Data

fatihcavdur@uludag.edu.tr

The Travelling Salesman Problem

c_{ij} = the distance between cities i and j

$f_t(i, S)$ = the minimum distance that must be traveled to complete a tour if the $t-1$ cities in the set S have been visited and city i was the last city visited

For $t = 4$,

$$f_4(2, \{2,3,4\}) = c_{21} = 1{,}334$$

$$f_4(3, \{2,3,4\}) = c_{31} = 1{,}559$$

$$f_4(4, \{2,3,4\}) = c_{41} = 809$$

For $t = 1,2,3$,

$$f_t(i, S) = \min_{j \notin S \wedge j \neq 1} \{c_{ij} + f_{t+1}(j, S \cup \{j\})\}; \quad t = 1,2,3$$

fatihcavdur@uludag.edu.tr

The Travelling Salesman Problem

Stage 3 Computations:

$$f_3(2,\{2,3\}) = c_{24} + f_4(4,\{2,3,4\}) = 1{,}397 + 809 = 2{,}206$$

$$f_3(3,\{2,3\}) = c_{34} + f_4(4,\{2,3,4\}) = 921 + 809 = 1{,}730$$

$$f_3(2,\{2,4\}) = c_{23} + f_4(3,\{2,3,4\}) = 1{,}343 + 1{,}559 = 2{,}902$$

$$f_3(4,\{2,4\}) = c_{43} + f_4(3,\{2,3,4\}) = 921 + 1{,}559 = 2{,}480$$

$$f_3(3,\{3,4\}) = c_{32} + f_4(2,\{2,3,4\}) = 1{,}343 + 1{,}334 = 2{,}677$$

$$f_3(4,\{3,4\}) = c_{42} + f_4(2,\{2,3,4\}) = 1{,}397 + 1{,}334 = 2{,}731$$

fatihcavdur@uludag.edu.tr

The Travelling Salesman Problem

Stage 2 Computations:

$$f_2(2,\{2\}) = \min\begin{Bmatrix} c_{23} + f_3(3,\{2,3\}) = 1{,}343 + 1{,}730 = 3{,}073 \\ c_{24} + f_3(4,\{2,4\}) = 1{,}397 + 2{,}480 = 3{,}877 \end{Bmatrix}$$

$$f_2(3,\{3\}) = \min\begin{Bmatrix} c_{34} + f_3(4,\{3,4\}) = 921 + 2{,}731 = 3{,}652 \\ c_{32} + f_3(2,\{2,3\}) = 1{,}343 + 2{,}206 = 3{,}549 \end{Bmatrix}$$

$$f_2(4,\{4\}) = \min\begin{Bmatrix} c_{42} + f_3(2,\{2,4\}) = 1{,}397 + 2{,}902 = 4{,}299 \\ c_{43} + f_3(3,\{3,4\}) = 921 + 2{,}677 = 3{,}598 \end{Bmatrix}$$

The Travelling Salesman Problem

Stage 1 Computations:

$$f_1(1,\{\cdot\}) = \min\begin{Bmatrix} c_{12} + f_2(2,\{2\}) = 1{,}334 + 3{,}073 = 4{,}407 \\ c_{13} + f_2(3,\{3\}) = 1{,}559 + 3{,}549 = 5{,}108 \\ c_{14} + f_2(4,\{4\}) = 809 + 3{,}598 = 4{,}407 \end{Bmatrix}$$

The (alternative) optimal tour is City 1 (NY) – City 4 (Chicago) – City 3 (Dallas) – City 2 (Miami) with length of $f_1(1,\{\cdot\}) = 4{,}407$.

fatihcavdur@uludag.edu.tr

The End

fatihcavdur@uludag.edu.tr