

---

**Homework 6**

JingXuan Yang, SZ160310217

December 20, 2019

---

- 6.1. A college student has 7 days remaining before final examinations begin in her four courses, and she wants to allocate this study time as effectively as possible. She needs at least 1 day on each course, and she likes to concentrate on just one course each day, so she wants to allocate 1, 2, 3, or 4 days to each course. Having recently taken an OR course, she decides to use dynamic programming to make these allocations to maximize the total grade points to be obtained from the four courses. She estimates that the alternative allocations for each course would yield the number of grade points shown in the following table:

Study Days	Estimated Grade Points			
	Course			
	1	2	3	4
1	3	5	2	6
2	5	5	4	7
3	6	6	7	9
4	7	9	8	9

Solve this problem by dynamic programming.

*Solution.* A dynamic programming formulation for this problem is

$$\begin{cases} \text{Stage } n = \text{course } n \\ x_n = \text{study days for course } n \\ \text{State } s_n = \text{number of study days still available to remaining courses} \end{cases}$$

Let  $p_i(x_i)$  be the estimated grade points from allocating  $x_i$  study days to course  $i$ . Then the objective function is

Maximize

$$Z = \sum_{i=1}^4 p_i(x_i),$$

subject to

$$\sum_{i=1}^4 x_i = 7,$$

$x_i$  are positive integers,  $i = 1, 2, 3, 4$ .

The recursive relationship for this problem is

$$f_n^*(s_n) = \max_{x_n} \{p_n(x_n) + f_{n+1}^*(s_n - x_n)\}, \quad \text{for } n = 1, 2, 3.$$

For the last stage ( $n = 4$ ),

$$f_4^*(s_4) = \max_{x_4} \{p_4(x_4)\}.$$

Beginning with the last stage ( $n = 4$ ),

$$x_4^* = s_4, \quad f_4^*(s_4) = p_4(s_4),$$

which is shown in Tab.1.

Table 1: Dynamic programming for  $n = 4$

$s_4$	$f_4^*(s_4)$	$x_4^*$
1	6	1
2	7	2
3	9	3
4	9	4

We now move backward to start from the next-to-last stage  $n = 3$ , which is shown in Tab.2.

Table 2: Dynamic programming for  $n = 3$

$f_3(s_3, x_3) = p_3(x_3) + f_4^*(s_3 - x_3)$						
$s_3 \backslash x_3$	1	2	3	4	$f_3^*(s_3)$	$x_3^*$
2	8				8	1
3	9	10			10	2
4	11	11	13		13	3
5	11	13	14	14	14	3 or 4

We go on moving to the former stage  $n = 2$ , which is shown in Tab.3.

Table 3: Dynamic programming for  $n = 2$

$f_2(s_2, x_2) = p_2(x_2) + f_3^*(s_2 - x_2)$						
$s_2 \backslash x_2$	1	2	3	4	$f_2^*(s_2)$	$x_2^*$
3	13				13	1
4	15	13			15	1
5	18	15	14		18	1
6	19	18	16	17	19	1

We now are ready to move backward to solve the original problem where we are starting from stage ( $n = 1$ ), which is shown in Tab.4.

Table 4: Dynamic programming for  $n = 1$

$f_1(s_1, x_1) = p_1(x_1) + f_2^*(s_1 - x_1)$						
$s_1 \backslash x_1$	1	2	3	4	$f_1^*(s_1)$	$x_1^*$
7	21	23	21	20	23	2

Thus the optimal solution has  $x_1^* = 2$ , which makes  $s_2 = 5$ , so  $x_2^* = 1$ . Then  $s_3 = 4$ , so  $x_3^* = 3$ , which makes  $s_4 = x_4^* = 1$ . Since  $f_1^* = 23$ , this (2, 1, 3, 1) allocation of study days will yield an estimated total of 23 grade points.

- 6.2. A company will soon be introducing a new product into a very competitive market and is currently planning its marketing strategy. The decision has been made to introduce the product in three phases. Phase 1 will feature making a special introductory offer of the product to the public at a greatly reduced price to attract first-time buyers. Phase 2 will involve an intensive advertising campaign to persuade these first-time buyers to continue purchasing the product at a regular price. It is known that another company will be introducing a new competitive product at about the time that phase 2 will end. Therefore, phase 3 will involve a follow-up advertising and promotion campaign to try to keep the regular purchasers from switching to the competitive product.

A total of \$4 million has been budgeted for this marketing campaign. The problem now is to determine how to allocate this money most effectively to the three phases. Let  $m$  denote the initial share of the market (expressed as a percentage) attained in phase 1,  $f_2$  the fraction of this market share that is retained in phase 2, and  $f_3$  the fraction of the remaining market share that is retained in phase 3. Use dynamic programming to determine how to allocate the \$4 million to maximize the final share of the market for the new product, i.e., to maximize  $mf_2f_3$ .

Assume that *any* amount within the total budget can be spent in each phase, where the estimated effect of spending an amount  $x_i$  (in units of millions of dollars) in phase  $i$  ( $i = 1, 2, 3$ ) is

$$\begin{cases} m = 10x_1 - x_1^2 \\ f_2 = 0.40 + 0.10x_2 \\ f_3 = 0.60 + 0.07x_3 \end{cases}$$

[Hint: After solving for the  $f_2^*(s)$  and  $f_3^*(s)$  functions analytically, solve for  $x_1^*$  graphically.]

*Solution.* A dynamic programming formulation for this problem is

$$\begin{cases} \text{Stage } n = \text{phase } n \\ x_n = \text{money allocated to phase } n \\ \text{State } s_n = \text{amount of money still available to remaining phases} \end{cases}$$

The objective function of this problem is

Maximize

$$Z = mf_2f_3 = (10x_1 - x_1^2)(0.40 + 0.10x_2)(0.60 + 0.07x_3),$$

subject to

$$\begin{aligned} \sum_{i=1}^3 x_i &= 4, \\ x_i &\geq 0, \quad i = 1, 2, 3. \end{aligned}$$

Let  $f_1 = m$ , the recursive relationship is

$$f_n^*(s_n) = \max_{0 \leq x_n \leq s_n} \{f_n(x_n) \cdot f_{n+1}^*(s_n - x_n)\}, \quad \text{for } n = 1, 2.$$

For  $n = 3$ ,

$$f_3^*(s_3) = \max_{0 \leq x_3 \leq s_3} \{f_3(x_3)\}$$

Beginning with the last stage ( $n = 3$ ), since  $f_3(s_3) = 0.60 + 0.07s_3$  is a monotonically increasing function of  $s_3$ ,  $x_3 = s_3$  is the desired maximizing value. The corresponding table is shown in Tab.5.

Table 5: Dynamic programming for  $n = 3$ 

$s_3$	$f_3^*(s_3)$	$x_3^*$
$0 \leq s_3 \leq 4$	$0.60 + 0.70s_3$	$s_3$

We now move backward to start from the next-to-last stage  $n = 2$ ,

$$\begin{aligned} f_2(s_2, x_2) &= (0.40 + 0.10x_2)(0.60 + 0.07s_3) \\ &= (0.40 + 0.10x_2)[0.60 + 0.07(s_2 - x_2)] \end{aligned}$$

Let

$$\begin{aligned} \frac{\partial}{\partial x_2} f_2(s_2, x_2) &= 0.10[0.60 + 0.07(s_2 - x_2)] - 0.07(0.40 + 0.10x_2) \\ &= 0.06 + 0.007(s_2 - x_2) - 0.028 - 0.007x_2 \\ &= 0.032 + 0.007s_2 - 0.014x_2 \\ &= 0, \end{aligned}$$

which yields

$$x_2^* = \frac{0.032 + 0.007s_2}{0.014} = \frac{32 + 7s_2}{14}.$$

Let

$$s_2 < \frac{32 + 7s_2}{14},$$

which yields

$$s_2 < \frac{32}{7}.$$

Since  $0 \leq s_2 \leq 4$ , the above inequality always holds. Therefore,

$$\frac{\partial}{\partial x_2} f_2(s_2, x_2) > 0, \quad \text{for } 0 \leq x_2 \leq s_2,$$

so that  $x_2 = s_2$  is the desired maximizing value. The above analysis is shown in Tab.6.

Table 6: Dynamic programming for  $n = 2$ 

$s_2$	$f_2^*(s_2)$	$x_2^*$
$0 \leq s_2 \leq 4$	$0.24 + 0.06s_2$	$s_2$

For the first-stage problem ( $n = 1$ ),

$$\begin{aligned} f_1(4, x_1) &= (10x_1 - x_1^2)f_2^*(s_2) \\ &= (10x_1 - x_1^2)f_2^*(4 - x_1) \\ &= (10x_1 - x_1^2)[0.24 + 0.06(4 - x_1)] \\ &= (10x_1 - x_1^2)(0.48 - 0.06x_1) \end{aligned}$$

Let

$$\begin{aligned} \frac{\partial}{\partial x_1} f_1(4, x_1) &= (10 - 2x_1)(0.48 - 0.06x_1) - 0.06(10x_1 - x_1^2) \\ &= 4.8 - 0.6x_1 - 0.96x_1 + 0.12x_1^2 - 0.6x_1 + 0.06x_1^2 \\ &= 4.8 - 2.16x_1 + 0.18x_1^2 \\ &= 0, \end{aligned}$$

which yields

$$x_1^* = \frac{2.16 \pm \sqrt{2.16^2 - 4 \times 0.18 \times 4.8}}{2 \times 0.18} = 2.945 \text{ or } 9.055$$

The graph of  $f_1(4, x_1)$  is shown in Fig.1. Since  $0 < 2.945 < 4 < 9.055$ ,  $x_1 = 2.945$  is the desired maximizing value, with  $Z = f_1^*(4, 2.945) = 6.302$ .

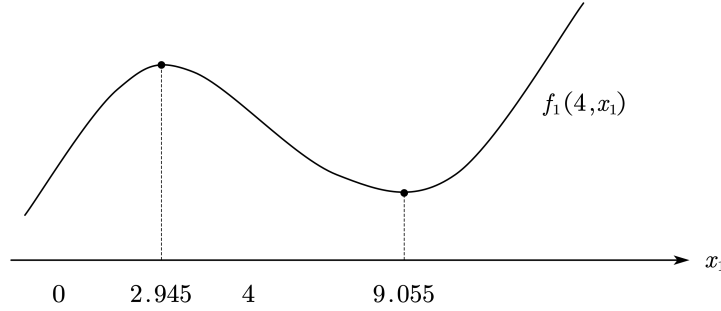


Figure 1: Graph of  $f_1(4, x_1)$

The results are summarized as follows:

Table 7: Dynamic programming for  $n = 1$

$s_1$	$f_1^*(s_1)$	$x_1^*$
4	6.302	2.945

Therefore, by tracing back through the tables for  $n = 2$ ,  $n = 3$ , respectively, and setting  $s_n = x_{n-1}^*$  each time, the resulting optimal solution is  $x_1^* = 2.945$ ,  $x_2^* = 1.055$ ,  $x_3^* = 0$ , with a total estimated market share of 6.302%.

- 6.3. Imagine that you have \$5000 to invest and that you will have an opportunity to invest that amount in either of two investments ( $A$  or  $B$ ) at the beginning of each of the next 3 years. Both investments have uncertain returns. For investment  $A$  you will either lose your money entirely or (with higher probability) get back \$10000 (a profit of \$5000) at the end of the year. For investment  $B$  you will get back either just your \$5000 or (with low probability) \$10000 at the end of the year. The probabilities for these events are as follows:

Investment	Amount Returned (\$)	Probability
$A$	0	0.3
	10000	0.7
$B$	5000	0.9
	10000	0.1

You are allowed to make at most *one* investment each year, and you can invest only \$5,000 each time. (Any additional money accumulated is left idle.)

Use dynamic programming to find the investment policy that maximizes the expected amount of money you will have after 3 years.

*Solution.* A dynamic programming formulation for this problem is

$$\begin{cases} \text{Stage } n = n\text{th year} \\ x_n = \text{investment made at stage } n, x_n \in \{0, A, B\} \\ \text{State } s_n = \text{amount of money in hand to begin stage } n \end{cases}$$

Since the objective of this problem is to maximize the expected amount of money after 3 years,

$$\begin{cases} f_n(s_n, x_n) = \text{total expected amount of money for stages } n, \dots, 3 \\ \quad \text{if system starts in state } s_n \text{ at stage } n, \text{ immediate decision is } x_n, \\ \quad \text{and optimal decisions are made thereafter,} \\ f_n^*(s_n) = \max_{x_n} \{f_n(s_n, x_n)\} \end{cases}$$

Since one can not invest less than \$5000, for  $0 \leq s_n < 5000$ ,

$$f_n(s_n, x_n) = f_{n+1}^*(s_n), \quad x_n^* = 0$$

For  $s_n \geq 5000$ ,

$$f_n(s_n, x_n) = \begin{cases} f_{n+1}^*(s_n), & \text{if } x = 0 \\ 0.3f_{n+1}^*(s_n - 5000) + 0.7f_{n+1}^*(s_n + 5000), & \text{if } x = A \\ 0.9f_{n+1}^*(s_n) + 0.1f_{n+1}^*(s_n + 5000), & \text{if } x = B \end{cases}$$

Beginning with the last stage ( $n = 3$ ), the calculations are shown in Tab.8.

Table 8: Dynamic programming for  $n = 3$

		$f_3(s_3, x_3)$			$f_3^*(s_3)$	$x_3^*$
$s_3$	$x_3$	0	A	B		
$0 \leq s_3 < 5000$	$s_3$	$s_3$			$s_3$	0
$s_3 \geq 5000$	$s_3$	$s_3 + 2000$	$s_3 + 500$		$s_3 + 2000$	A

We now move backward to start from the next-to-last stage  $n = 2$ , the results obtained are shown in Tab.9.

Table 9: Dynamic programming for  $n = 2$

		$f_2(s_2, x_2)$			$f_2^*(s_2)$	$x_2^*$
$s_2$	$x_2$	0	A	B		
$0 \leq s_2 < 5000$	$s_2$	$s_2$			$s_2$	0
$5000 \leq s_2 < 10000$	$s_2$	$s_2 + 3400$	$s_2 + 2500$		$s_2 + 3400$	A
$s_2 \geq 10000$	$s_2$	$s_2 + 4000$	$s_2 + 2500$		$s_2 + 4000$	A

We now are ready to move backward to solve the original problem where we are starting from stage ( $n = 1$ ), which is shown in Tab.10.

Table 10: Dynamic programming for  $n = 1$ 

		$f_1(s_1, x_1)$			$f_1^*(s_1)$	$x_1^*$
$s_1 \backslash x_1$		0	A	B		
	5000	8400	9800	8960	9800	A

Therefore, the optimal investment policy is to invest in  $A$  if the money in hand is greater than or equal to \$5000, with a total expected amount of money after 3 years of \$9800.