12

Dynamic Programming

"Repetition is the only form of permanence that nature can achieve"

12 : 1. INTRODUCTION

Many decision-making problems involve a process that takes place in several stages (multi-stage process) in such a way that at each stage, the process is dependent on the strategy chosen. Such type of problems are called *Dynamic Programming Problems (D.P.P.)*. Thus dynamic programming is concerned with the theory of multi-stage decision process, *i.e.*, the process in which a sequence of inter-related decisions has to be made. Mathematically, a D.P.P. is a decision-making problem in n-variables, the problem being subdivided into n sub-problems (segments) each sub-problem being a decision-making problem in one variable only. The solution to a D.P.P. is achieved sequentially starting from one (initial) stage to the next till the final stage is reached.

Principle of Optimality

It may be interesting to note that the concept of dynamic programming is largely based upon the *principle of optimality* due to Bellman, *viz.*,

"An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

The principle of optimality implies that given the initial state of a system, an optimal policy for the subsequent stages does not depend upon the policy adopted at the preceding stages. That is, the effect of a *current* policy decision on any of the policy decisions of the preceding stages need not be taken into account at all. It is usually referred to as the *Markovian property* of dynamic programming.

12 : 2. THE RECURSIVE EQUATION APPROACH

There exist different approaches to solve a D.P.P. To illustrate the one called '*recursive equation approach*' we consider the following decision-making situation :

There are n machines each of which can perform two different kinds of work. If z machines work on the first kind of work, commodities worth $g(z)$ are produced and if they work on the second kind of work, commodities worth $h(z)$ are produced. It is also known to the company that after completing a job, some machines get partly inoperative. For the first kind of work, $a(z)$ machines are left over whereas for the second kind of work, $b(z)$ are left over. If k jobs are to be performed what policy should be adopted for producing goods in such a way that the total value of the goods produced is maximized ?

We subdivide this problem into several decision-making problems. Each such problem is an allocation problem of allocating a job to the optimum number of machines. In the initial (first) stage all the n machines are available for use. Let x_1 be the number of machines assigned to job *1* and $n - x_1$ be the number of machines assigned to job 2. Clearly $0 \le x_1 \le n$. Let $f_N(n)$ denote the *total* optimal value of the produced goods when we start with n machines and work in N stages, at each stage determining a sequential optimum.

Thus

$$f_1(n) = \underset{0 \le x_1 \le n}{\text{Max.}} \; [g(x_1) + h\,(y_1)] \qquad \text{where } y_1 = n - x_1.$$

Now consider stage 2. The number of machines that are now available in order, is $n_1 = a\,(x_1) + b(y_1)$. From the principle of optimality, given the current stage, the optimum alternative for the current stage is obtained by optimizing the sum of (a) the optimum value of the goods produced at all previously considered stages and (b) the value of goods produced at the current stage.

Thus, if x_2 machines are allotted a particular job and $n_1 - x_2$ some other one in stage 2, then the value of the goods produced in this stage is $g(x_2) + h(y_2)$ where $y_2 = n_1 - x_2$. The optimum allocation at stage 2, therefore, corresponds to

$$f_2(n) = \underset{0 \le x_1 \le n}{\text{Max.}} \; \{g\,(x_1) + h\,(y_1) + f_1\,[a\,(x_1) + b(y_1)]\},\; y_1 = n - x_1$$

or, in general,

$$f_2(n) = \underset{0 \le x_1 \le n}{\text{Max.}} \; \{[g(x) + h\,(y)] + f_1[a(x) + b(y)]\},\; y = n - x$$

By similar argument, we have

$$f_3(n) = \underset{0 \le x_1 \le n}{\text{Max.}} \; \{[g(x) + h\,(y)] + f_2\,[a\,(x) + b\,(y)]\},\; y = n - x$$

and $$f_k\,(n) = \underset{0 \le x_1 \le n}{\text{Max.}} \; \{[g(x) + h\,(y)] + f_{k-1}\,[a(x) + b\,(y)]\},\; y = n - x,\; k > 1$$

This is a recurrence equation for f_i's.

This recurrence relation connects the optimal decisions function for the N-stage problem with the optimal decision function for the $(N - 1)$ stage sub-problem.

12 : 3. CHARACTERISTICS OF DYNAMIC PROGRAMMING

The basic features which characterize the dynamic programming problem are as follows :

(a) The problem can be sub-divided into stages with a policy decision required at each stage. A stage is a device to sequence the decisions. That is, it decomposes a problem into sub-problems such that an optimal solution to the problem can be obtained from the optimal solutions to the sub-problems.

(*b*) Every stage consists of a number of states associated with it. The states are the different possible conditions in which the system may find itself at that stage of the problem.

(*c*) Decision at each stage converts the current stage into state associated with the next stage.

(*d*) The state of the system at a stage is described by a set of variables, called *state variables.*

(*e*) when the current state is known, an optimal policy for the remaining stages is independent of the policy of the previous ones.

(*f*) To identify the optimum policy for each state of the system, a recursive equation is formulated with n stages remaining, given the optimal policy for each state with $(n-1)$ stages left.

(*g*) Using recursive equation approach each time the solution procedure moves backward stage by stage for obtaining the optimum policy of each state for that particular stage, till it attains the optimum policy beginning at the initial stage.

12 : 4. DYNAMIC PROGRAMMING ALGORITHM

The computational procedure for solving a problem by dynamic programming approach can be summarized in the following steps :

Step 1. Identify the decision variables and specify objective function to be optimized under certain limitations, if any.

Step 2. Decompose (or divide) the given problem into a number of smaller sub-problems (or stages). Identify the state variables at each stage and write down the transformation function as a function of the state variable and decision variables at the next stage.

Step 3. Write down a general recursive relationship for computing the optimal policy. Decide whether forward or backward method is to follow to solve the problem.

Step 4. Construct appropriate stages to show the required values of the return function at each stage.

Step 5. Determine the overall optimal policy or decisions and its value at each stage. There may be more than one such optimal policy.

Remarks. 1. Generally the solution of a recursive equation involves two types of computations, according as the system is continuous or discrete. In the first case the optimal decision at each stage is obtained by using the usual classical methods of optimization. In the second case, a tabular computational scheme is followed.

2. If the dynamic programming problem is solved by using the recursive equation starting from the first through the last stage, *i.e.*, obtaining the sequence $f_1 \to f_2 \to \ldots \to f_N$, the computation involved is called the *forward* computational procedure. If the recursive equation is formulated in a different way so as to obtain the sequence $f_N \to f_{N-1} \to \dots \to f_1$, then the computation is known as the *backward* computational procedure.

SAMPLE PROBLEM

1201. (*Optimal Sub-division Problem*). *Divide a positive quantity c into n parts in such a way that their products is a maximum.*

Or

Maximize $z = y_1 . y_2 \dots\dots y_n$ *subject to the constraints :*

$$y_1 + y_2 + \dots\dots + y_n = c \text{ and } y_j \geq 0;\ j = 1, 2, \dots, n.$$

[*Marathwada M.Sc. (Appl. Math.) 1982; Meerut M.Sc. (Math.) 1983; Ranchi M.Sc. (Math.) 1982; Kurukshetra M.Sc. (Math.) 1982; Madurai B.Sc. (Appl. Math.) 1982; Delhi M.Sc. (O.R.) 1985*]

Solution. Let y_j be the j^{th} part of the positive quantity c ($j = 1, 2, \dots, n$), then each j corresponding to part y_j may be regarded as a stage. Now, since y_j may assume any non-negative value satisfying the constraint

$$y_1 + y_2 + \dots + y_n = c,$$

the alternatives at each stage are infinite. This means that y_j may be considered to be continuous.

Let $f_n(c)$ denote the maximum attainable product. Clearly, this will be a function of both n and c.

If we regard c as a fixed quantity and n as the number of stages, which varies over positive integers, then a recursive equation connecting $f_n(c)$ and $f_{n-1}(c)$ is

$$f_n(c) = \max_{0 < x \leq c} \{ x . f_{n-1}(c - x) \}$$

For $n = 1$ (*i.e.*, one stage problem), we write

$$f_1(c) = \max_{y_1 = c} \{y_1\} = c \quad \text{(initially true)}$$

For $n = 2$ (*i.e.*, two stage problem), the quantity c is divided into two parts, say $y_1 = x$ and $y_2 = c - x$. Then

$$\begin{aligned} f_2(c) &= \max . \{y_1 . y_2\} \\ &= \max_{0 < x \leq c} \{x . (c - x)\} \\ &= \max_{0 < x \leq c} \{x . f(c - x)\}, \text{ since } f_1(c - x) = c - x. \end{aligned}$$

Similarly, for $n = 3$, the quantity c is divided into three parts, given the initial choices of x which leaves $c - x$ to be further divided into two parts. Denote the maximum possible product for $(c - x)$ into two parts by $f_2(c - x)$. Then using the principle of optimality, we have

$$f_3(c) = \max_{0 < x \leq c} \{x . f_2(c - x)\}.$$

Continuing in a similar manner, the recursive equation for general value of x is given by

$$f_n(c) = \max_{0 \leq x \leq c} \{x . f_{n-1}(c - x)\}$$

We now solve the recurrence equation formulated above.

For $n = 2$, the function $c . (c - x)$ attains its maximum value at $x = c/2$ satisfying the condition $0 < x \leq c$. Thus

$$f_2(c) = \frac{c}{2}\left(c - \frac{c}{2}\right) = \left(\frac{c}{2}\right)^2$$

$\therefore$ The optimal policy is $(c/2,\ c/2)$ and $f_2(c) = (c/2)^2$.

For $n = 3$, $f_3(c) = \max\limits_{0 \le x \le c} \left\{ x \cdot \left(\frac{c-x}{2}\right)^2 \right\}$, since $f_2(c-x) = \left(\frac{c-x}{2}\right)^2$.

Now, since the maximum value of $x\left(\frac{c-x}{2}\right)^2$ is attained for $x = c/3$ satisfying the condition $0 < x \le c$; therefore

$$f_3(c) = \left\{ \frac{c}{3} \cdot \frac{1}{4}\left(c - \frac{c}{3}\right)^3 \right\} = \left(\frac{c}{3}\right)^3$$

Thus, for $n = 3$, we have

Optimal policy : $\left(\frac{c}{3},\ \frac{c}{3},\ \frac{c}{3}\right)$ and $f_3(c) = \left(\frac{c}{3}\right)^3$

In general, for n-stage problem, we assume that

Optimal policy : $\left(\frac{c}{n},\ \frac{c}{n},\ \dots,\ \frac{c}{n}\right)$ and $f_n(c) = \left(\frac{c}{n}\right)^n$,

for $n = 1, 2, \dots, m$.

Now, for $n = m+1$, the recursive equation is

$$f_{m+1}(c) = \max_{0 < x \le c} \{x \cdot f_m(c - x)\} = \max_{0 < x \le c} \left\{ x \cdot \left(\frac{c-x}{m}\right)^m \right\}$$

$$= \left(\frac{c}{m+1}\right)^{m+1},$$

as the maximum value of $x\left(\frac{c-x}{m}\right)^m$ is attained at $x = \frac{c}{m+1}$, *i.e.*, the result is also true for $n = m+1$.

Hence, by mathematical induction, the optimal policy is

$$\left(\frac{c}{n},\ \frac{c}{n},\ \dots\dots,\ \frac{c}{n}\right) \text{ and } f_n^{\circ}(c) = \left(\frac{c}{n}\right)^n. \square$$

1202. *Use dynamic programming to show that*

$$z = p_1 \log p_1 + p_2 \log p_2 + \dots + p_n \log p_n \quad \textit{subject to the constraints :}$$

$$p_1 + p_2 + \dots + p_n = 1 \quad \text{and} \quad p_j \ge 0 \quad (j = 1, 2, \dots, n)$$

is minimum when $p_1 = p_2 = \dots = p_n = 1/n$.

[Rohilkhand M.Sc. (Math.) 1983; Nagarjuna M.Sc. (Stat.) 1989; Meerut M.Sc. (Math.) 1985]

Solution. The problem here is to divide unity into n parts so as to minimize the quantity $\sum p_i \log p_i$.

Let $f_n(1)$ denote the minimum attainable sum of $p_i \log p_i$ $(i = 1, 2, \dots, n)$.

For $n = 1$ (stage *1*), we have

$$f_1(1) = \min_{0 < x \le 1} \{p_1 \log p_1\} = 1 \log 1,$$

as unity is divided only into $p_1 = 1$ part.

For $n = 2$, the unity is divided into two parts p_1 and p_2, such that $p_1 + p_2 = 1$.

If $p_1 = x$ and $p_2 = 1 - x$, then

$$f_2(1) = \min_{0<x\leq 1} \{p_1 \log p_1 + p_2 \log p_2\}$$
$$= \min_{0<x\leq 1} \{x \log x + (1-x) \log (1-x)\}$$
$$= \min_{0<x\leq 1} \{x \log x + f_1 (1-x)\}$$

In general, for an n-stage problem, the recursive equation is

$$f_n(1) = \min_{0<x\leq 1} \{p_1 \log p_1 + p_2 \log p_2 + \ldots\ldots + p_n \log p_n\}$$
$$= \min_{0<x\leq 1} \{x \log x + f_{n-1} (1-x)\}$$

We now solve this recursive equation.

For $n = 2$ (stage 2), the function $x \log x + (1-x) \log (1-x)$ attains its minimum value at $x = 1/2$ satisfying the condition $0 < x \leq 1$. Thus,

$$f_2(1) = \frac{1}{2} \log \frac{1}{2} + \left(1 - \frac{1}{2}\right) \log \left(1 - \frac{1}{2}\right) = 2 \left(\frac{1}{2} \log \frac{1}{2}\right).$$

Similarly, for stage 3, the minimum value of the recursive equation is obtained as

$$f_3(1) = \min_{0<x\leq 1} \{x \log x + f_2 (1-x)\}$$
$$= \min_{0<x\leq 1} \left\{x \log x + 2 \left(\frac{1-x}{2}\right) \log \left(\frac{1-x}{2}\right)\right\}$$

Now, since the minimum value of

$$x \log x + 2 \left(\frac{1-x}{2}\right) \log \left(\frac{1-x}{2}\right)$$

is attained at $x = 1/3$ satisfying $0 < x \leq 1$, we have

$$f_3(1) = \frac{1}{3} \log \frac{1}{3} + 2 \left(\frac{1}{3} \log \frac{1}{3}\right) = 3 \left(\frac{1}{3} \log \frac{1}{3}\right).$$

$\therefore$ Optimal policy is : $p_1 = p_2 = p_3 = 1/3$.

In general, for n-stage problem we assume that

Optimal policy : $p_1 = p_2 = \ldots = p_n = \frac{1}{n}$ and $f_n(1) = n \left\{\frac{1}{n} \log \frac{1}{n}\right\}$.

This can be shown easily using mathematical induction.

For $n = m + 1$, the recursive equation is

$$f_{m+1}(1) = \min_{0<x\leq 1} \{x \log x + f_m (1-x)\}$$
$$= \min_{0<x\leq 1} \left[x \log x + m \left\{\frac{1-x}{m} \log \left(\frac{1-x}{m}\right)\right\}\right]$$
$$= \frac{1}{m+1} \log \frac{1}{m+1} + m \left\{\frac{1}{m+1} \log \frac{1}{m+1}\right\}$$
$$= (m+1) \left\{\frac{1}{m+1} \log \frac{1}{m+1}\right\},$$

since minimum of $x \log x + \frac{1-x}{m} \log \frac{1-x}{m}$ is attained at $x = \frac{1}{m+1}$.

Hence the required optimal policy is

$$\left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right) \text{ with } f_n^\circ(1) = n\left(\frac{1}{n}\log\frac{1}{n}\right). \square$$

1203. *Use dynamic programming to solve the following problem :*

Minimize $z = y_1^2 + y_2^2 + y_3^2$ *subject to the constraints :*

$$y_1 + y_2 + y_3 \geq 15 \text{ and } y_1, y_2, y_3 \geq 0.$$

[*Madras B.E. 1982; Meerut M.Sc. (Math.) 1988*]

Solution. Since the decision variables are y_1, y_2 and y_3, the given problem is a three stage problem defined as follows :

$s_3 = y_1 + y_2 + y_3 \geq 15$, $s_2 = y_1 + y_2 = s_3 - y_3$ and $s_1 = y_1 = s_2 - y_2$

Therefore the functional (recurrence) relation is

$$f_1(s_1) = \min_{0 \leq y_1 \leq s_1} y_1^2 = (s_2 - y_2)^2$$

$$f_2(s_2) = \min_{0 \leq y_2 \leq s_2} \{y_1^2 + y_2^2\} = \min_{0 \leq y_2 \leq s_2} \{y_2^2 + f_1(s_1)\},$$

and
$$f_3(s_3) = \min_{0 \leq y_3 \leq s_3} \{y_1^2 + y_2^2 + y_3^2\} = \min_{0 \leq y_3 \leq s_3} \{y_3^2 + f_2(s_2)\}$$

$$\therefore \quad f_2(s_2) = \min_{0 \leq y_2 \leq s_2} \{y_2^2 + (s_2 - y_2)^2\}$$

$$= \left(\frac{1}{2}s_2\right)^2 + \left(s_2 - \frac{1}{2}s_2\right)^2 = \frac{1}{2}s_2^2;$$

since the function $y_2^2 + (s_2 - y_2)^2$ attains its minimum value at $y_2 = \frac{1}{2}s_2$.

Again, $f_3(s_3) = \min\limits_{0 \leq y_3 \leq s_3} \{y_3^2 + f_2(s_2)\} = \min\limits_{0 \leq y_3 \leq s_3} \{y_3^2 + \frac{1}{2}(s_3 - y_3)^2\}$

or $f_3(15) = \min\limits_{0 \leq y_3 \leq 15} \{y_3^2 + \frac{1}{2}(15 - y_3)^2\}$, since $s_3(y_1 + y_2 + y_3) \geq 15$.

Since the minimum value of the function $y_3^2 + \frac{1}{2}(15 - y_3)^2$ occurs at $y_3 = 5$; we have

$$f_3(15) = \{5^2 + \tfrac{1}{2}(15 - 5)^2\} = 75$$

Thus $s_3 = 15$ implies that $y_3^\circ = 5$;

$s_2 = s_3 - y_3 = 15 - 5 = 10$ implies that $y_2^\circ = \frac{1}{2}s_2 = 5$

$s_1 = s_2 - y_2 = 10 - 5 = 5$ implies that $y_1^\circ = s_1 = 5$.

Hence the optimal policy is

$$(5, 5, 5) \text{ with } f_3^\circ(15) = 75. \square$$

PROBLEMS

Use dynamic programming to find the value of

1204. Maximum $z = y_1 \cdot y_2 \cdot y_3$ subject to the constraints :

$$y_1 + y_2 + y_3 = 5, \; y_1, y_2, y_3 \geq 0.$$

[*Ranchi M.Sc. (Stat.) 1983*]

1205. Minimum $z = y_1 + y_2 + \ldots + y_n$ subject to the constraints :

$$y_1 \cdot y_2 \ldots\ldots y_n = d, \; y_j \geq 0 \; ; j = 1, 2, \ldots, n.$$

[*Calicut M.Sc. (Stat.) 1979; Karnataka B.E. (Mech.) 1984; P.S.G. Coimbatore M.Sc. (Math.) 1992*]

[*Hint*. For $n = 2$, the recurrence relation is

$$f_2(d) = \min_{0 \le y \le d} \{y + d/y\}$$ with d as the initial decision].

1206. Minimum $z = y_1^2 + y_2^2 + \dots y_n^2$ subject to the constraints :

$$y_1 . y_2 . \dots . y_n = c, \; y_1, y_2, \dots, y_n \ge 0.$$

[*Meerut M.Sc. (Math.) 1992*]

1207. Maximum $z = b_1x_1 + b_2x_2 + \dots + b_nx_n$ subject to the constraints :

$$x_1 + x_2 + \dots + x_n = c$$

$$x_1, x_2, x_3, \dots, x_n \ge 0.$$

[*Cochin M.Sc. (Math.) 1985; Meerut M.Sc. (Math.) 1990*]

1208. Illustrate the dynamic programming approach by solving the following problem :

Maximize $12x_1^2 + 27x_2^2 + 147x_3^2$ subject to the constraints :

$$x_1 + x_2 + x_3 = 1, \; x_1, x_2, x_3 \ge 0.$$

1209. Find the minimum value of $x_1^2 + 2x_2^2 + 4x_3$ subject to the constraints :

$$x_1 + 2x_2 + x_3 \ge 8 \quad x_1, x_2, x_3 \ge 0.$$

[*Meerut M.Sc. (Math.) 1989*]

1210. Find the maximum value of $z = x_1^2 + 2x_2^2 + 4x_3$ subject to the constraints :

$$x_1 + 2x_2 + x_3 \le 8, x_1, x_2, x_3 \ge 0.$$

1211. Find the maximum value of $z = -x_1^2 - 2x_2^2 + 3x_2 + x_3$ subject to the conditions :

$$x_1 + x_2 + x_3 \le 1, \quad x_1, x_2, x_3 \ge 0.$$

1212. Obtain the functional equation for maximizing $z = g_1(x_1) + g_2(x_2) \dots + g_n(x_n)$ subject to the constraints :

$$x_1 + x_2 + \dots + x_n = c, \; x_j \ge 0, \; j = 1, 2, \dots, n.$$

[*Delhi M.Sc. (Math.) 1975*]

1213. Develop the functional equation to determine $m_1, m_2, m_3, \dots\dots m_n$ so as to maximize

$$z = \sum_{i=1}^{n} m_i \left(\frac{p_i}{m_n}\right)^{\alpha},$$

subject to the constraints :

$$m_1 + m_2 + m_3 + \dots m_n = M \quad \text{and} \quad m_j \ge 0 \; (j = 1, 2, \dots, n).$$

[*Delhi M.Sc. (Math.) 1979*]

12 : 5. SOLUTION OF DISCRETE D.P.P.

Many problems such as production allocation, long-term planning, equipment replacement, multi-stage chemical processes, etc. can be solved by Dynamic Programming, using convenient tabular computations. This is best illustrated with the help of some sample problems.

SAMPLE PROBLEMS

1214 (***Product Allocation Problem***). *The owner of a chain of four grocery stores has purchased six crates of fresh strawberries. The estimated probability distribution of potential sales of the strawberries before spoilage differ among the four stores. The following table gives the estimated total expected profit at each store, when it is allocated various number of crates :*

Number of crates	1	2	3	4
0	0	0	0	0
1	4	2	6	2
2	6	4	8	3
3	7	6	8	4
4	7	8	8	4
5	7	9	8	4
6	7	10	8	4

For administrative reasons, the owner does not wish to split crates between stores. However, he is willing to distribute zero crates to any of his stores.

Find the allocation of six crates to four stores so as to maximize the expected profit. [*Ranchi B.I.T. B.Sc. (Prod. Engg.) 1984; Nagarjuna B. Tech. 1985; I.I.I.E. (Grad.) 1981; Ravi Shankar B.E. (Mech.) 1980*]

Solution. Let the four stores be considered as four stages in a dynamic programming formulation. The decision variables x_j ($j = 1, 2, 3, 4$) denote the number of crates allocated as the jth stage from the previous one.

Now let $P_j(x_j)$ be the expected profit from allocation of x_j crates to store j. Then the problem can be formulated as an L.P.P. as follows :

Maximize $z = P_1(x_1) + P_2(x_2) + P_3(x_3) + P_4(x_4)$ subject to the constraints :

$$x_1 + x_2 + x_3 + x_4 = 6, \quad x_1, x_2, x_3, x_4 \geq 0.$$

Let there be s crates available for j remaining stores and x_j be the initial allocation. Define $f_j(x_j)$ as the value of the optimal allocation for stores 1 through 4 both inclusive. Thus for stage $j = 1$,

$$f_1(s, x_1) = \{P_1(x_1)\}$$

If $f_j(s, x_j)$ be the profit associated with the optimum solution $f_j^*(s)$ ($j = 1, 2, 3, 4$), then

$$f_1^*(s) = \max_{0 \leq x_1 \leq s} P_1(x_1).$$

Thus the recurrence relation is

$$f_j(s, x_j) = P_j(x_j) + f_{j+1}^*(s - x_j) \qquad \text{for } j = 1, 2, 3, 4$$

and

$$f_j^*(s) = \max_{0 \leq x_j \leq s} \{P_j(x_j) + f_{j+1}^*(s - x_j)\}.$$

The solution to this problem starts with $f_4^*(s)$ and is completed when $f_1^*(s)$ is obtained.

The computations for one stage problem (*i.e.,* for $j = 1$) are as follows :.

s	$f_4^*(s)$	x_4^*
0	0	0
1	2	1
2	3	2
3	4	3
4	4	3, 4
5	4	3, 4, 5
6	4	3, 4, 5, 6

For $j = 2$, we have a two-stage problem. The computations are as follows :

$$f_3(s\ ,\ x_3)\ =\ P_3(x_3) + f_3^*(s - x_3)$$ *Optimum Sol.*

s \ x_2	0	1	2	3	4	5	6	$f_3^*(s)$	x_3^*
0	0 + 0							0	0
1	0 + 2	6 + 0						6	1
2	0 + 3	6 + 2	8 + 0					8	1, 2
3	0 + 4	6 + 3	8 + 2	8 + 0				10	2
4	0 + 4	6 + 4	8 + 3	8 + 2	8 + 0			11	2
5	0 + 4	6 + 4	8 + 4	8 + 3	8 + 2	8 + 0		12	2
6	0 + 4	6 + 4	8 + 4	8 + 4	8 + 3	8 + 2	8 + 0	12	2, 3

For $j = 3$, we have three stage problem. So, we have

$$f_2(s, x_2)\ =\ P_2(x_2) + f_3^*(s - x_2)$$ *Optimum Sol.*

s \ x_2	0	1	2	3	4	5	6	$f_2^*(s)$	x_2^*
0	0 + 0							0	0
1	0 + 6	2 + 0						6	0
2	0 + 8	2 + 6	4 + 0					8	0, 1
3	0 + 10	2 + 8	4 + 6	6 + 0				10	0, 1, 2
4	0 + 11	2 + 10	4 + 8	6 + 6	8 + 0			12	1, 2, 3
5	0 + 12	2 + 11	4 + 10	6 + 8	8 + 6	9 + 0		14	2, 3, 4
6	0 + 12	2 + 12	4 + 11	6 + 10	8 + 8	9 + 6	10 + 0	16	3, 4

For $j = 4$, we have the required four-stage problem :

$$f_1(s, x_1)\ =\ P_1\ (x_1) + f_2^*(s - x_1)$$ *Optimum Sol.*

s \ x_1	0	1	2	3	4	5	6	$f_2^*(s)$	x_1^*
6	0 + 16	4 + 14	6 + 12	7 + 10	7 + 8	7 + 6	7 + 0	18	1, 2

From above computations it is observed that the maximum profit of Rs. 18 can be obtained by choosing the following eight alternative solutions :

Store 1 x_1^*	Store 2 x_2^*	Store 3 x_3^*	Store 4 x_4^*
1	2	2	1
1	3	1	1
1	3	2	0
1	4	1	0
2	1	2	1
2	2	1	1
2	2	2	0
2	3	1	0

□

1215 *(Cargo Load Problem). A vessel is to be loaded with stocks of 3 items. Each unit of item i has a weight w_i and value r_i. The maximum cargo weight the vessel can take is 5 and the details of the three items are as follows :*

i	w_i	r_i
1	1	30
2	3	80
3	2	65

Develop the recursive equation for the above case and find the most valuable cargo load without exceeding the maximum cargo weight by using dynamic programming. **[Nagarjuna B.Tech. (July) 1985]**

Solution. We have to determine how many units of three items are to be loaded. So it is a three-stage problem. Let x_j (j = ', 2, 3). denote the three decisions. Let $f_j(x_j)$ denote the value of the optimal allocation for the three types of items.

If $f_j(s, x_j)$ be the value associated with the optimum solution $f_j^*(s)$, ($j = 1, 2, \ldots, n$) then we have

$$f_1^*(s) = \max_{0 \le x_1 \le s} f_1(s, x_j)$$

and
$$f_j^*(s) = \max_{0 \le x_j \le s} \{ p_j(x_j) + f_{j+1}^*(s - x_j) \}, \quad j = 1, 2, 3$$

where $P_j(x_j)$ denotes the expected value obtained from allocation of x_j units of weight to the item j.

Now, for one stage problem (*i.e.*, for one item cargo loading)

$$f_1^*(s) = \max_{x_1} \{30\, x_1\},$$

where the largest value of x_1 is $[W/w_1] = [5/1] = 5$.

We have the following tabular computations :

Value of $30\,x_1$ — *Optimum Solution*

s \ x_1	0	1	2	3	4	5	$f_1^*(s)$	x_1^*
0	0						0	0
1	0	30					30	1
2	0	30	60				60	2
3	0	30	60	90			90	3
4	0	30	60	90	120		120	4
5	0	30	60	90	120	150	150	5

For 2-stage problem, the largest value of x_2 is $[5/3] = 1$ and

$$f_2^*(x) = \max_{x_2} \{ 80x_2 + f_1(s - 3x_2) \}$$

Value of $80x_2 + f_1^*(s - 3x_2)$ — *Optimum Solution*

s \ x_2	0	1	$f_2^*(s)$	x_2^*
0	0 + 0 = 0		0	0
1	0 + 30 = 30		30	0
2	0 + 60 = 60		60	0
3	0 + 90 = 90	80 + 0 = 80	90	0
4	0 + 120 = 120	80 + 30 = 110	120	0
5	0 + 150 = 150	80 + 60 = 140	150	0

For 3-stage problem, the largest value of x_3 is $[5/3] = 2$ and

$$f_3^*(s) = \max_{x_3} \{ 65\,x_3 + f_2^*(s - 2x_3) \}$$

Value of $65x_3 + f_2^*(s - 2x_3)$ — *Optimum Solution*

s \ x_2	0	1	2	$f_3^*(s)$	x_3^*
0	0 + 0			0	0
1	0 + 30			30	0
2	0 + 60	65 + 0 = 65		65	1
3	0 + 90	65 + 30 = 95		95	1
4	0 + 120	65 + 60 = 125	130 + 0 = 130	130	2
5	0 + 150	65 + 90 = 155	130 + 30 = 160	160	2

Given $W = 5$, the optimum solution, therefore, is

$$x_3^* = 2,\; x_2^* = 0 \text{ and } x_1^* = 1$$

with $f_3^*(s) = 160$ = maximum value of load. □

PROBLEMS

1216. In 18th century, when transportation systems were not developed, a family wanted to travel home to reach a friend's house in other part of the country. But they had a choice of various routes and haltages in between from their home to final destination. Cost of travel from each point to the other points en route, based on relevant factors such as distance, difficulties, mode of available transportation etc. are given below :

	2	3	4
1	7	5	4

	5	6	7
2	8	3	9
3	10	7	6
4	4	5	6

	8	9
4	6	8
6	7	4
7	3	6

	10
8	5
9	4

Find the most safest route of travelling so that the total travelling cost becomes minimum. *[Ranchi B.I.T. B.Sc. (Prod. Engg.) 1984]*

1217. A member of a certain political party is making plans for an upcoming presidential election. He has received the services of six volunteer workers for precinct work and he wishes to assign them to three precincts in such a way as to maximize their effectiveness. He feels that it would be inefficient to assign a worker to more than one precinct, but he is willing to assign no workers to anyone of the precincts if they can accomplish more in other precincts.

The following table gives the estimated increase in the plurality of the party's candidate if it were allocated various number of workers :

Number of workers	*Precinct* 1	2	3
0	0	0	0
1	25	20	33
2	42	38	43
3	55	54	47
4	63	65	50
5	69	73	52
6	74	80	53

How many of the workers should be assigned to each of the three precincts in order to maximize total estimated increase in the plurality of the party's candidate ?
[I.I.I.E. (Grad.) 1981; Delhi M.Sc. (Math) 1973]

1218. The World Health Council is devoted to improving health care in the under-developed countries of the world. It now has five *medical teams* available to allocate among three such countries to improve their medical care, health education and training programmes. Therefore, the council needs to determine how may teams (if any) to allocate to each of these countries to maximize the total effectiveness of the five teams. The measure of effectiveness being used is *additional man-years of life*. (For a particular country, this measure equals the country's increased life expectancy in years times its population). The following table gives the estimated additional man-years of life (in multiple of 1,000) for each country for each possible allocation of medical teams.

No. of medical teams	Thousands of additional man-years of life Country 1	2	3
0	0	0	0
1	45	20	50
2	70	45	70
3	90	75	90
4	105	110	100
5	120	150	130

Determine how many teams to be allocated to each country for maximum effectiveness. Also form the recursive equation.

[*Madras B.E. (Prod.) 1980; Madurai M.Sc. (Appl. Sc.) 1980*]

1219. A Government space project is conducting research on a certain engineering problem that must be solved before man can fly to moon safely.

Three research teams are currently trying three different approaches for solving this problem. The estimate has been made that under present circumstances, the probability that the respective teams— call them *A*, *B* and *C*— will not succeed are 0·40, 0·60 and 0·80 respectively. Thus the current probability that all three teams will fail is (0·40) (0·60) (0·80) = 0·192. Since the objective is to minimise this probability, the decision has been made to assign two or more top scientists among the three teams in order to lower it as much as possible.

The following table gives the estimated probability that the respective teams will fail when 0, 1 or 2 additional scientists are added to that team :

Number of new scientists		Team *A*	*B*	*C*
	0	0·40	0·60	0·80
	1	0·20	0·40	0·50
	2	0·15	0·20	0·30

How should the additional scientists be allocated to the team ?

[*Delhi M.Sc. (Math.) 1979, 1983*]

1220. A ship is to be loaded with stock of 3 items. Each unit of item 'n' has a weight w_n and value v_n. The maximum cargo weight the ship can take is 5 and the details of the three items are as follows :

Item (n)	*Weight* (w_n)	*Value* (v_n)
1	2	7
2	3	10
3	1	3

Find the most valuable cargo load without exceeding the maximum cargo weight by using dynamic programming.

1221. A truck can carry a total of 10 tons of a product. Three types of product are available for shipment. Their weights and values are tabulated. Assuming that at least one of each type must be shipped, determine the loading which will maximize the total value :

Type	*Value (in Rs.)*	*Weight (tons)*
A	20	1
B	50	2
C	60	2

12 : 6. SOLUTION OF L.P.P. BY DYNAMIC PROGRAMMING

Consider the general linear programming problem :

Maximize $z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$

subject to the constraints :

$$a_{i1}x_1 + a_{i2}x_2 + \ldots\ldots + a_{in}x_n \le b_i \ ; \ i = 1, 2, \ldots, m$$

$$\text{and } x_j \ge 0 \ ; \ j = 1, 2, \ldots, n.$$

This problem can be formulated as a dynamic programming problem as follows :

Let the general linear programming problem be considered as a multi-stage problem with each activity j ($j = 1, 2, \ldots, n$) as individual stage. Then, this is a n-stage problem and the decision variables (alternatives) are the levels of activities x_j (≥ 0) at stage j. As x_j is continuous, each activity has an infinite number of alternatives within the feasible region.

We know that allocation problems are the particular types of linear programming problems. These problems require the allocation of available resources to the activities. Each constraint represents the limitation of different resources and $b_1, b_2, \ldots, b_m$ are the amounts of available resources. Since there are m resources, the state must be represented by an m-component vector s = (b_1, $b_2, \ldots, b_m$).

Let $f_n(b_1, b_2, \ldots, b_m)$ be the maximum value of the general linear programming problem defined above for stages $x_1, x_2, \ldots, x_n$ for states $b_1, b_2, \ldots, b_m$.

Using forward computational procedure, the recursive equation is :

$$f_j(b_1, b_2, \ldots, b_m) = \underset{0 \le x_j \le b}{\text{Max.}} \{c_j x_j + f_{j-1}(b_1 - a_{1j}x_j, b_2 - a_{2j}x_j, \ldots, b_m - a_{mj}x_j)\}$$

The maximum value of b that x_j can assume is

$$b = \text{Min.} \left\{ \frac{b_1}{a_{1j}}, \frac{b_2}{a_{2j}}, \ldots, \frac{b_m}{a_{mj}} \right\}$$

because the minimum value satisfies the set of constraints simultaneously.

SAMPLE PROBLEM

1222. *Use dynamic programming to solve the following L.P.P. :*

Maximize $z = 3x_1 + 5x_2$ *subject to the constraints :*

$$x_1 \le 4, \ x_2 \le 6, \ 3x_1 + 2x_2 \le 18 \ \text{ and } x_1, x_2 \ge 0.$$

[***Madurai*** ***B.Sc.*** ***(Appl. Math.) 1980; Meerut M.Sc. (Math.) 1987; Sambalpur M.Sc. (Math.) 1986***]

Solution. The problem consists of three resources and two decision variables. The states of the equivalent dynamic programming, therefore, are $b_1 = 4$, $b_2 = 6$ and $b_3 = 18$.

$\therefore$ For the first stage, we have

$$f_1(b_1, b_2, b_3) = \underset{0 \le x_1 \le b}{\text{Max.}} \{3x_1\}.$$

$$\therefore\ f_1(4, 6, 18) = \underset{0 \le x_1 \le 4}{\text{Max.}} \{3x_1\} = 3 \text{ Min.} \left\{ 4, \frac{18 - 2x_2}{3} \right\}$$

where $x_1^\circ = \text{Min.} \left\{ 4, \frac{18 - 2x_2}{3} \right\} = 4.$

For two-stage problem, the recursive equation is

$$f_2(4, 6, 18) = \underset{0 \le x_2 \le 6}{\text{Max.}} \left\{ 5x_2 + 3 \text{ Min.} \left(4, \frac{18 - 2x_2}{3} \right) \right\}$$

where $b = \text{Min.}\ (6, 18/2) = 6.$

$$\text{Now,} \quad \text{Min.} \left(4, \frac{18 - 2x_2}{3} \right) = \begin{cases} 4 & \text{if } 0 \le x_2 \le 3 \\ \dfrac{18 - 2x_2}{3}, & \text{if } 3 < x_2 \le 6 \end{cases}$$

$$\therefore\ 5x_2 + 3 \text{ Min.} \left(4, \frac{18 - 2x_2}{3} \right) = \begin{cases} 5x_2 + 12, & \text{if } 0 \le x_2 \le 3 \\ 18 + 3x_2, & \text{if } 3 < 3x_2 \le 6 \end{cases}$$

Since the maximum value of $5x_2 + 12$ is 27 at $x_2 = 3$ and maximum value of $18 + 3x_2$ is 36 at $x_2 = 6$; the optimum value of $f_2(4, 6, 18)$ will be 36.

Hence, the optimum solution is

Maximum $z = 36$ with $x_2^\circ = 6$ and $x_1^\circ = \text{Min.} \left\{ 4, \frac{18 - 2x_2}{3} \right\} = 2.$ □

PROBLEMS

Use dynamic programming to solve the following linear programming problems :

1223. Maximize $z = 3x_1 + 7x_2$ subject to the constraints :

$x_1 + 4x_2 \le 8,\ x_2 \le 2,\ x_1, x_2 \ge 0.$ **[*Delhi M.Sc. (Math.) 1983*]**

1224. Maximize $z = 8x_1 + 7x_2$ subject to the constraints :

$2x_1 + x_2 \le 8,\ 5x_1 + 2x_2 \le 15.$

$x_1 \ge 0$ and $x_2 \ge 0.$

[*I.I.Sc. (Ind. Man.) 1977; Cochin Dip. Oper. & Comp. Appl. 1981*]

1225. Maximize $z = 2x_1 + 5x_2$ subject to the constraints :

$2x_1 + x_2 \le 43,\ 2x_2 \le 46,\ x_1, x_2 \ge 0.$

[*P.S.G. Coimbatore M.Sc. (Math.) 1992; Bharthiar M.Sc. (Math.) 1986*]

1226. Solve the following linear programming problem by applying dynamic programming procedures. Explain the assumptions you make :

Maximize $z = 2x_1 + 4x_2$ subject to the constraints :

$2x_1 + 3x_2 \le 48,\ x_1 + 3x_2 \le 42,\ x_1 + x_2 \le 21,\ x_1, x_2 \ge 0.$

[*B.I.T. Ranchi B.E. (Prod.) 1984*]

12:7. SOME APPLICATIONS

SAMPLE PROBLEMS

1227. *Suppose there are n machines which can do two jobs. If x of them do the first job, then they produce goods worth $g(x) = 3x$ and if y of the machines do the second job then they produce goods worth $h(y) = 2{\cdot}5y$. The machines are subject to depreciation, so that after doing the first job only $a(x) = x/3$ machines remain available and after doing the second job only $b(y) = 2y/3$ machines remain available in the beginning of the second year.*