For $n = 3$, d is factorized into three factors y_1, y_2, and y_3.

If $y_1 = z$ and $y_2 y_3 = d/z$, then

$$f_3(d) = \min \{y_1 + y_2 + y_3\} = \min_{0 \le z \le d} \{ z + f_2(d/z) \}$$

Proceeding likewise, the recurrence relation for $n = i$ becomes

$$f_i(d) = \min_{0 \le z \le d} \{ z + f_{i-1}(d/z) \}, i = 2, \ldots, n.$$

Now proceed to solve this functional equation as follows :

$$f_1(d) = d, f_2(d) = \min_{0 \le z \le d} \{ z + d/z \} = \sqrt{d} + d/\sqrt{d} = 2\sqrt{d} \qquad \text{(by calculus method)}$$

$$f_3(d) = \min_{0 \le z \le d} \{ z + f_2(d/z) \} = \min_{0 \le z \le d} \{ z + 2\sqrt{(d/z)} \} = d^{1/3} + 2\sqrt{(d/d^{1/3})} = 3d^{1/3} \text{ and so on.}$$

By induction hypothesis, assume for $n = m$

$$f_m(d) = m d^{1/m}.$$

Now, the result can be proved for $n = m + 1$ as follows :

$$f_{m+1}(d) = \min_{0 \le z \le d} \{ z + f_m(d/z) \} = \min_{0 \le z \le d} \{ z + m\sqrt{(d/z)^{1/m}} \}$$

$$= (m+1)\, d^{1/(m+1)} \qquad \text{(by calculus method)}$$

Hence the optimal policy will be

$$(d^{1/n}, d^{1/n}, \ldots, d^{1/n}) \text{ with } f_n(d) = n d^{1/n}.$$

Example 17. *Solve the following problem using dynamic programming :*

Minimize $\mathbf{z} = y_1^2 + y_2^2 + \ldots + y_n^2$, *subject to the constraints*

$y_1 y_2 y_3 \ldots y_n = b$, *and* $y_1, y_2, y_3, \ldots, y_n \ge 0$.

Solution. Let $f_n(b)$ be the minimum attainable sum of given n terms.

For $n = 1$,
$$f_1(b) = \min_{z=b} \{z^2\} = b^2 \qquad \ldots(1)$$

For $n = 2$, let $y_1 = z$, $y_2 = b/z$. Then

$$f_2(b) = \min \{y_1^2 + y_2^2\} = \min_{0 \le z \le b} \{z^2 + f(b/z)^2\}. \qquad \ldots(2a)$$

Since $f_1(b) = b^2$, therefore $f_1(b/z) = (b/z)^2$. Consequently,

$$f_2(b) = \min_{0 \le z \le b} \{z^2 + f_1(b/z)\}. \qquad \ldots(2b)$$

Similarly, for $n = 3$,
$$f_3(b) = \min_{0 \le z \le b} \{z^2 + f_2(b/z)\}. \qquad \ldots(3)$$

by using the principle of optimality.

Thus, the functional equation for this problem becomes

$$f_n(b) = \min_{0 \le z \le b} \{z^2 + f_{n-1}(b/z)\}. \qquad \ldots(4)$$

To find the optimal policy **:**

From eqn. (2a),
$$f_2(b) = \min_{0 \le z \le b} \{z^2 + (b/z)^2\}.$$

Take $F(z) = z^2 + (b/z)^2$, then $\frac{dF}{dz} = 2z - \frac{2b^2}{z^3} = 0$ [for maximum or minimum of $F(z)$]

which gives $z = b^{1/2}$, $y_1 = b^{1/2}$, $y_2 = b/z = b^{1/2}$.

Since $\frac{d^2F}{dz^2}$ is + ve, indicating $F(z)$ is minimum.

Also, $f_2(b) = (b^{1/2})^2 + (b^{1/2})^2 = 2b$.

Hence, optimal policy is : $(b^{1/2}, b^{1/2})$; $f_2(b) = 2b$.

Again, from (3) $\qquad f_3(b) = \min\limits_{0 \le z \le b} \{z^2 + f_2(b/z)\}$

Since $f_2(b) = 2b \Rightarrow f_2(b/z) = 2\,(b/z)$, therefore

$$f_3(b) = \min_{0 \le z \le b} \{z^2 + 2\,(b/z)\}.$$

Let $F(c) = z^2 + 2\,(b/z)$, and proceed as earlier to obtain the minimum of $F(z)$ for $z = b^{1/3}$.

Therefore, $\qquad f_3(b) = (b^{1/3})^2 + 2\,\dfrac{b}{b^{1/2}} = 3\,.\,b^{2/3} = (b^{1/3})^2 + (b^{1/3})^2 + (b^{1/3})^2$

which indicates $y_1 = y_2 = y_3 = b^{1/3}$.

Hence, optimal policy is $\qquad (b^{1/3}, b^{1/3}, b^{1/3})$; $f_3(b) = 3b^{2/3}$.

Continueing in this manner, the optimal policy for general n will be obtained as

$$(b^{1/n}, b^{1/n}, \ldots, b^{1/n}), \text{ and } F_n(b) = (b^{1/n})^2 + (b^{1/n})^2 + \ldots + n \text{ times} = n\,b^{2/n}.$$

Example 18 (Discrete Variables). *Solve the following problem using dynamic programming.*

Maximize $z = y_1^2 + y_2^2 + y_3^2$, *subject to* $y_1\, y_2\, y_3 \le 4$, *where* y_1, y_2, y_3 *are positive integers.*

[JNTU (Mech. & Prod.) 2004]

Solution. First, define *state variables* as

$$s_3 = y_1 y_2 y_3 \le 4,\quad s_2 = s_3/y_3 = y_1 y_2,\quad s_1 = s_2/y_2 = y_1$$

and proceeding exactly as ***Example 11*** to obtain the solution from the following tables.

Stage returns : $f_j(y_j) = y_j^2$, $j = 1, 2, 3$

y_j :	1	2	3	4
$f_j(y_j)$:	1	4	9	16

Stage transformations : $s_{j-1} = s_j/y_j$, $j = 2, 3$

s_j \ y_j	1	2	3	4
1	1	—	—	—
2	2	1	—	—
3	3	—	1	—
4	4	2	—	1

Recursive Operations

s_1	$F_1(s_1)$
1	1
2	4
3	9
4	16

	$f_2(y_2)$				$F_1(s_1) = f_1(y_1)$				$F_2(s_2)$
s_2 \ y_2	1	2	3	4	1	2	3	4	
1	1	—	—	—	1	—	—	—	2
2	1	4	—	—	4	1	—	—	5
3	1	—	9	—	9	—	1	—	10
4	1	4	—	16	16	4	—	1	17

	$f_3(y_3)$				$F_2(s_2)$				$F_3(s_3)$
s_3 \ y_3	1	2	3	4	1	2	3	4	
1	1	—	—	—	2	—	—	—	3
2	1	4	—	—	5	2	—	—	6
3	1	—	9	—	10	—	2	—	11
4	1	4	—	16	17	5	—	—	18

Thus, the required solution is given by : $y_3 = 1$, $y_2 = 1$, $y_1 = 4$, max $z = 18$.

EXAMINATION PROBLEMS

1. Find the minimum $z = x_1 + x_2 + x_3 + \dots + x_n$, when $x_1 x_2 x_3 \dots x_n = d$, and $x_1, x_2, x_3, \dots, x_n \geq 0$.
2. Use the principle of optimality to solve the problem :

$$\text{Minimize } z = \sum_{j=1}^{N} x_j^{\alpha} \text{ subject to } x_1 x_2 x_3 \dots x_N = r,\ x_i \geq 1,\ i = 1, 2, \dots, N,$$

where $r \geq 1$ and $\alpha > 0$ are given fixed numbers.

33.9. MODEL V : SYSTEM INVOLVING MORE THAN ONE CONSTRAINT

Dynamic programming models discussed so far involve only one constraint apart from non-negativity conditions. In fact, the dynamic programming method can be applied to problems involving more than one constraint also. In single constraint problems, there has to be single state variable for each stage, while in multi-constraint problems there has to be one state variable per constraint per stage. The structure of problems is of such type that sometimes it is possible to reduce the number of state variables. The stage transformation becomes more and more complicated with the increase in number of constraints and consequently the state variables. Large number of constraints can almost be a forbidding computational burden on the dynamic programming method. Fundamental concepts of the procedure will remain the same.

Example 19. *Maximize* $z = y_1^3 + y_2^3 + y_3^3$, *subject to the constraints* $y_1 + y_2 + y_3 \leq 6$, $y_1 y_2 y_3 \leq 6$, *where* y_1, y_2 and y_3 *are positive integers.*

Solution. First define two sets of stage variables as follows :

$$s_3 = y_1 + y_2 + y_3 \qquad t_3 = y_1 y_2 y_3$$
$$s_2 = s_3 - y_3 = y_1 + y_2 \qquad t_2 = t_3/y_3 = y_1 y_2$$
$$s_1 = s_2 - y_2 = y_1 \qquad t_1 = t_2/y_2 = y_1$$

Obviously, feasible values of y_i are 1, 2, 3 and 4.

For stage $j = 1$, stage transformations will give the following possible values of s_1 and t_1.

y_1	s_1	t_1
1	1	1
2	2	2
3	3	3
4	4	4

For $j = 2, 3,$ following table gives the transformations :

$$s_{j-1} = T_{j-1}(s_j, y_j),\ t_{j-1} = T_{j-1}(t_j, y_j)$$

(s_j, t_j) \ y_j	1	2	3	4
	(s_{j-1}, t_{j-1})			
(1, 1)	(0, 1)	(–,–)	(–,–)	(–,–)
(2, 2)	(1, 2)	(0, 1)	(–,–)	(–,–)
(3, 3)	(2, 3)	(1,–)	(0, 1)	(–,–)
(4, 4)	(3, 4)	(2, 2)	(1,–)	(–,–)
(5, 5)	(4, 5)	(3,–)	(2,–)	(0, 1)
(6, 6)	(5, 6)	(4, 3)	(3, 2)	(1,–)

In order to preserve the validity of constraints it is not necessary to consider s_j, $t_j > 6$. Since fractional and negative integral values are not considered, so these are denoted by dash (–) in above table.

Optimizations

Stage 1. $F_1(s_1, t_1) = y_1^3$

y_1	s_1	t_1	$F_1(s_1, t_1)$
1	1	1	1
2	2	2	8
3	3	3	27
4	4	4	64

Stage 2. $F_2(s_2, t_2) = \max_{y_2} [y_2^3 + F_1(s_1, t_1)]$

y_2	s_1	t_1	$F_1(s_1, t_1)$	$y_2^3 + F_1(s_1, t_1)$	s_2	t_2	$f_2(s_2, t_2)$
1	1	1	1	2	2	1	2
	2	2	8	9	3	2	9
	3	3	27	28	4	3	28
	4	4	64	65	5	4	65
2	1	1	1	9	3	2	×
	2	2	8	16	4	4	16
	3	3	27	35	5	6	35
3	1	1	1	28	4	3	×
	2	2	8	35	5	6	×
4	1	1	1	65	5	4	×

Stage 3. $F_3(s_3, t_3) = \max_{y_3} [y_3^3 + F_2(s_2, t_2)]$

y_3	s_2	t_2	$F_2(s_2, t_2)$	$y_3^3 + F_2(s_2, t_2)$	s_3	t_3	$F_3(s_3, t_3)$
1	2	1	2	3	3	1	3
	3	2	9	10	4	2	10
	4	3	28	29	5	3	29
	4	4	16	17	5	4	17
	5	5	65	66	6	4	66
	5	6	35	36	6	6	36
2	2	1	2	10	4	2	×
	3	2	9	17	5	4	×
	4	3	28	36	6	6	×
3	2	1	2	29	5	3	×
	3	2	9	36	6	6	×
4*	2	1	2	66*	6	4	66*

Now, proceeding in the backward direction, optimal decisions are

$$(y_1, y_2, y_3) = (4, 1, 1) \text{ or } (1, 1, 4) \text{ or } (1, 4, 1).$$

Hence, max $F_3(s_3, t_3) = 66$ for $(s_3, t_3) = (6, 4)$.

Now we shall give a mathematical formulation of *general (multistage)* dynamic programming problem.

33.10. MATHEMATICAL FORMULATION OF MULTISTAGE MODEL

Let there be a system in an initial *state* described by a vector $\mathbf{s}_N$. As a result of certain decisions denoted by the vector $\mathbf{d}$, this system finally reaches the state $\mathbf{s}_0$ as shown in Fig. 33.5. The rectangle represents the *transformation* T_N functionally as

$$\mathbf{s}_0 = T_N(\mathbf{s}_N, \mathbf{d}) \quad ...(33.16)$$

and $\mathbf{s}_N$ is regarded as *input* and $\mathbf{s}_0$ as the *output.*

Suppose a real valued function

$$\psi_N (\mathbf{s}_N, \mathbf{d}) \quad ...(33.17)$$

called the *objective* or *the return function*, is associated with the transformation T_N.

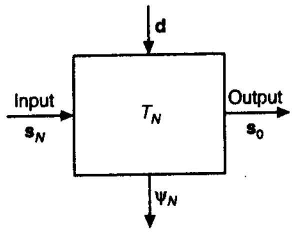

Fig. 33.5

The objective is to determine a given input $\mathbf{s}_N$ to optimize (*minimize* or *maximize*) ψ_N subject to the constraint (33.16). The transformation (33.16) is a constraint on $\mathbf{d}$ with prescribed values of $\mathbf{s}_N$ and $\mathbf{s}_0$.

If it is possible to decompose the problem into j number of stages, $1 \leq j \leq N$, then $\mathbf{s}_j$ will represent the input at the jth stage. Starting from the initial state $\mathbf{s}_N$, the system is considered to pass through the successive states

$s_{N-1}, s_{N-2}, \ldots, s_2, s_1$, before reaching the final state s_0. Thus each state s_{j-1} is the function of the input state s_j and the decision vector d_j, *i.e.*

$$s_{j-1} = T_j(s_j, d_j) \qquad \ldots(33.18)$$

It is also assumed that there exists a *stage return function*

$$f_j(s_j, d_j) \qquad \ldots(33.19)$$

at the *j*th stage. Also, the return function ψ_N is some function of stage returns, *i.e.*

$$\psi_N = \psi_N(f_N, f_{N-1}, \ldots, f_2, f_1) \qquad \ldots(33.20)$$

The return function can also be expressed in the form

$$\psi_N = \psi_N(s_N, d_N, d_{N-1}, \ldots, d_1) \qquad \ldots(33.21)$$

by virtue of (33.18) and (33.19). Clearly, (33.21) is equivalent to (33.17).

Now, the situation is diagrammatically explained in *Fig.* 33.6, which is the *serial multistage model.*

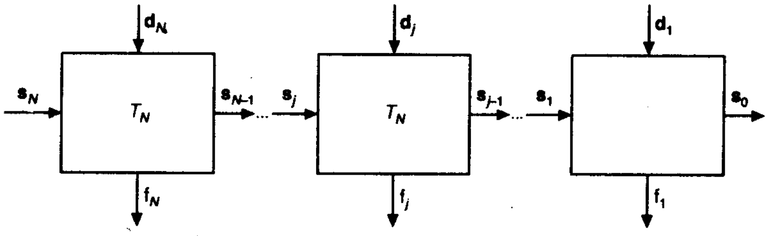

Fig. 33.6

It is concluded that under certain conditions the problem of optimizing $\psi_N(s_N, d)$ subject to $s_0 = T_N(s_N, d)$ can be transformed to a serial multistage problem of determining sequentially optimal decisions d_j^*, $1 \le j \le N$, which optimizes $\psi_N(f_N, f_{N-1}, \ldots, f_2, f_1)$.

From the examples discussed so far, it would seem to suggest that if ψ_N is of the form

$$\psi_N = f_N \, o \, f_{N-1} \, o \, f_{N-2} \, o \, f_2 \, o \, f_1 \qquad \ldots(33.22)$$

where o represents a *composition operator* indicating either *addition* or *multiplication,* then

$$\Psi_N = f_{N-1} \qquad \ldots(33.23)$$

where $$\psi_{N-1} = f_{N-1} \, o \, f_{N-2} \, o \ldots o \, f_2 \, o \, f_1 \qquad \ldots(33.24)$$

and then it may be possible to affirm positively that

$$F_N(s_N) = \max_{d_N, d_{N-1}, \ldots, d_2, d_1} \psi_N(s_N, d_N, d_{N-1}, \ldots, d_2, d_1) = \max_{d_N} [f_N \, o \, F_{N-1}(s_{N-1})] \qquad \ldots(33.25)$$

where $$F_{N-1}(s_{N-1}) = \max_{s_{N-1}, \ldots, s_2, s_1} \psi_{N-1}(s_{N-1}, d_{N-1}, \ldots, d_2, d_1) \qquad \ldots(33.26)$$

The improved form of the return function (33.23) is called *separability.* If it is possible to separate all $\psi_N, \psi_{N-1}, \ldots, \psi_2$ successively in this order, the recursive equation may be proposed,

$$F_j(s) = \max_{d_j} [f_j \, o \, F_{j-1}(s_{j-1})], \; 2 \le j \le N \qquad \ldots(33.27)$$

with $$F_1(s_1) = \max_{d_1} f_1 \qquad \ldots(33.28)$$

subject to $$s_{j-1} = T_j(s_j, d_j), \; 2 \le j \le N \qquad \ldots(33.29)$$

which *may* enable us to solve the maximum problem recursively.

Now, it is always a point of discussion whether this approach will always work or if not, what are the conditions under which it works ?

Following examples of failure will make the situation clear.

Counter Examples :

(i) Consider the function of the form $\psi_3 = f_3 f_2 + f_1$

Now, this function is not separable in the order 3, 2, 1 because no matter how one define $\psi_2(f_2, f_1)$. It is not possible to express ψ_3 as $f_3 \, o \, \psi_2$ where *o* denotes either addition or multiplication.

(ii) On the other hand, the function of the form $\psi_3 = f_3 + f_2 f_1$ is separable, because it is possible to define

$$\psi_2 = f_2 f_1, \psi_1 = f_1$$

and then $\psi_3 = f_3 + \psi_2$, $\psi_2 = f_2 \psi_1$.

(iii) As another example, the function of the form $\psi_4 = f_4 + f_3 f_2 + f_1$ is not separable in either direction.

Hence, all functions of the form

$$\psi_N = f_N \, o \, f_{N-1} \, o \, f_{N-2} \, o \, \ldots \, o \, f_2 \, o \, f_1$$

are not separable.

(iv) Maximize $\psi_3 = f_3 f_2 f_1$ where $f_3 = y_3$, $f_2 = y_2$, $f_1 = y_1$, subject to

$$1 \le y_1 \le 3, -2 \le y_2 \le -1, -1 \le y_3 \le 0,$$

The solution is : Max $\psi_3 = 6$.

Now adopting the dynamic programming approach :

$$\max \psi_3 = F_3 = \max_{y_3} (y_3 F_2), \text{ where } F_2 = \max_{y_2} (y_2 F_1), F_1 = f_1 = y_1.$$

Proceeding in backward direction

$$F_2 = \max_{y_2} (y_2 y_1) = -1, F_3 = \max_{y_3} (-y_3) = 1$$

which is wrong.

Hence, *the recursive optimization may not work, even though the function is separable.*

Q. Setup the recursive relation, using dynamic programming approach, when an N stage objective function is to be maximized. **[Meerut (M.Sc. Maths.) 90]**

33.11. DECOMPOSITION

Definition 1. An optimization problem is said to be ***decomposable*** if it can be solved by recursive optimization through N-stages, at each stage optimization being done over one decision variable. In other words, validity of recursive equation (33.27) implies decomposability.

The monotonicity of a function is also being defined for use in the subsequent discussion.

Definition 2. The function $f(x, y)$ is said to be **monotonic non-decreasing** function of x for all feasible values of y if :

$$x_1 > x_2 \Rightarrow f(x_1, y) \ge f(x_2, y)$$

for every feasible value of y. It is said to be **monotonic non-increasing** if :

$$x_1 > x_2 \Rightarrow f(x_1, y) \le f(x_2, y)$$

for every feasible value of y.

Theorem 33.1. *In a serial double-stage optimization problem if:*

(i) the objective function ψ_2 is a separable function of stage returns $f_1(\mathbf{s}_1, \mathbf{d}_1)$ and $f_2(\mathbf{s}_2, \mathbf{d}_2)$, and

(ii) ψ_2 is a monotonic non-decreasing function of f_1 for every feasible value of f_2,

then the problem is decomposable.

Proof. As discussed in ***Sec.* 7.10**, the objective function $\psi_2(f_2, f_1)$ is separable if $\psi_2 = f_2 \, o \, \psi_1$, $\psi_1 = f_1$.

Suppose this condition holds, and further ψ_2 is monotonic non-decreasing function of f_1 for feasible values of f_2.

The theorem is considered for the maximization case and similar treatment may be adopted for minimization also.

As introduced in ***Sec.* 33.10**, the equivalence of the following expressions is given by

$$F_2(\mathbf{s}_2) = \max_{\mathbf{d}_1, \mathbf{d}_2} \psi_2(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1) \quad \ldots(33.30)$$

$$= \max_{\mathbf{d}_1, \mathbf{d}_2} [f_2(\mathbf{s}_2, \mathbf{d}_2) \, o \, f_1(\mathbf{s}_1, \mathbf{d}_1)] \quad \ldots(33.31)$$

$$= \max_{\mathbf{d}_1, \mathbf{d}_2} [f_2(\mathbf{s}_2, \mathbf{d}_2) \; o \; f_1(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1)] \quad ...(33.32)$$

Using the transformation relation

$$\mathbf{s}_1 = F_2(\mathbf{s}_2, \mathbf{d}_2). \quad ...(33.33)$$

Also,

$$F_1(\mathbf{s}_1) = \max_{\mathbf{d}_1} f_1(\mathbf{s}_1, \mathbf{d}_1) = \max_{\mathbf{d}_1} f_1(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1). \quad ...(33.34)$$

Let

$$F_2{}^*(\mathbf{s}_2) = \max_{\mathbf{d}_2} = [f_2(\mathbf{s}_1, \mathbf{d}_1) \; o \; F_1(\mathbf{s}_1)] \quad ...(33.35)$$

$$= \max_{\mathbf{d}_2} [f_2(\mathbf{s}_2, \mathbf{d}_2) \; o \; \max_{\mathbf{d}_1} f_2(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1)] \quad ...(33.36)$$

Now comparing (33.32) and (33.36)

$$F_2{}^*(\mathbf{s}_3) \geq F_2(\mathbf{s}_2). \quad ...(33.37)$$

If

$$\max_{\mathbf{d}_1} f_1(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1) = f_1(\mathbf{s}_1, \mathbf{d}_2, \mathbf{d}_1{}^*), \quad ...(33.38)$$

then

$$f_1(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1) \leq f_1(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1{}^*).$$

Since ψ_2 is a monotonic non-decreasing function of f_1, inequality implies,

$$\psi_2(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1) \leq \psi_2(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1{}^*)$$

or

$$\psi_2(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1{}^*) \geq \max_{\mathbf{d}_1} \psi_2(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1) \quad ...(33.39)$$

Now, from (33.36) and (33.38)

$$F_2{}^*(\mathbf{s}_2) = \max_{\mathbf{d}_2} [f_2(\mathbf{s}_2, \mathbf{d}_2) \; o \; f_1(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1{}^*)]$$

$$= \max_{\mathbf{d}_2} \psi_2(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1{}^*), \quad \text{[from (33.30) and (33.39)]}$$

$$\geq \max_{\mathbf{d}_2} \max_{\mathbf{d}_1} \psi_2(\mathbf{s}_2, \mathbf{d}_2, \mathbf{d}_1) \quad \text{[from (33.39)]}$$

$$= F_2(\mathbf{s}_2) \quad \text{[from (33.30)]} \quad ...(33.40)$$

From (33.37) and (33.40)

$$F_2(\mathbf{s}_2) = F_2{}^*(\mathbf{s}_2) \text{ or } F_2(\mathbf{s}_2) = \max_{\mathbf{d}_2} [f_2 \; o \; F_1(\mathbf{s}_1)], \quad \text{[from (33.35)]}$$

which along with (33.34), are equations (33.27) and (33.28) for $N = 2$. Thus, by definition, the maximization problem is decomposable.

Hence, the theorem is proved.

The following theorem is a direct consequence of *Theorem* 33.1 and hence no further proof is needed. In fact, it is an extension to N-stage optimization problem.

Theorem 33.2. *If the real valued return function* $\psi_N(f_N, f_{N-1}, \dots, f_1)$ *satisfies :*

(i) The condition of separability, i.e.

$$\psi_N(f_N, f_{N-1}, \dots, f_1) = f_N \; o \; \psi_{N-1}$$

where $\psi_{N-1}(f_{N-1}, \dots, f_1)$ is real valued, and

(ii) ψ_N is monotonic non-decreasing function of ψ_{N-1} for every f_N, then ψ_N is decomposable, *i.e.*

$$\max_{\mathbf{d}_N, \dots, \mathbf{d}_1} \psi_N(f_N, \dots, f_1) = \max_{\mathbf{d}_N} [f_N \; o \max_{\mathbf{d}_{N-1}, \dots, \mathbf{d}_1} \psi_{N-1}]$$

Theorems 33.1 and 33.2 prove that the monotonicity is the *sufficient* condition for decomposability. To prove that it is *not* the necessary condition, following example, is sufficient.

Example. Maximize $\psi_2 = f_2 f_1$, where $f_1 = y_1, f_2 = y_2$, subject to $1 \leq y_1 \leq 4$; $-1 \leq y_2 \leq 1$.

Obviously, the solution is max $\psi_2 = 4$. Since ψ_2 decreases as f_1 increases for negative f_2, ψ_2 is not monotonic non-decreasing function for every value of f_2.

In order to show that the correct answer is obtained by dynamic programming approach.

$$\max_{y_2, y_1} (f_2, f_1) = \max_{y_2} (f_2, \max_{y_1} f_1) = \max_{y_2} (4f_2) = 4.$$

Q. 1. State a sufficient condition for a two-stage optimization problem to be solved by dynamic programming.

2. Deduce the dimensionality in dynamic programming. **[Delhi (OR) 93]**

3. Discuss the purchasing problem and prove the existence and uniqueness theorem.

33.12. BACKWARD AND FORWARD RECURSIVE APPROACH

The recursive approach in which $\mathbf{s}_j$ is the input and $\mathbf{s}_{j-1}$ is the output for the jth stage, where stage returns are expressed as functions of stage inputs, and the recursive analysis proceeds from stage 1 to stage N, is discussed earlier. This type of approach is called the *backward recursion* on account of stage transformation function being of the form $\mathbf{s}_{j-1} = T_j(\mathbf{s}_j, \mathbf{d}_j)$. The backward recursion can be conveniently used only when optimization with respect to a specific input $\mathbf{s}_N$ is needed, because in such a case the output $\mathbf{s}_0$ is not taken into account.

To optimize the system with respect to a prescribed output $\mathbf{s}_0$, it would be naturally convenient to reverse the direction. Treating $\mathbf{s}_j$ as the function of $\mathbf{s}_{j-1}$ and $\mathbf{d}_j$, and substitute $\mathbf{s}_j = Tj\ (\mathbf{s}_{j-1}, \mathbf{d}_j)$, $1 \le j \le N$, and also express stage returns as functions of stage output and then proceed from stage N to stage 1. Such an approach is called the ***forward recursive approach.***

In this case, input $\mathbf{s}_n$ and output $\mathbf{s}_0$ are prescribed parameters. Both of these parameters will be retained during analysis, and the optimal solution will then be a function of both the parameters. In multistage problems, there is no difference in applying these two approaches. Inputs and outputs both are fictitious concepts and are therefore interchangable. The problem can be solved in any direction by slightly modifying the notations.

In non-serial multistage systems which are important in automatic control systems, stages are not connected in series, but branches and loops may also occur therein. While dealing with such system by the dynamic programming approach, the difference of *forward* and *backward* recursion procedures becomes much considerable. Therefore, the forward recursion formulae are given.

Let the return function $\psi_1 = (\mathbf{s}_N, \mathbf{s}_0, \mathbf{d}_N, \ldots, \mathbf{d}_1)$ be a function of stage returns $f_j = (\mathbf{s}_j, \mathbf{s}_{j-1}, \mathbf{d}_j)$ in the form

$$\psi_1 = f_N \, o \, f_{N-1} \ldots o \, f_2 \, o \, f_1.$$

assuming the stage transformation function as

$$\mathbf{s}_j = T_j(\mathbf{s}_{j-1}, \mathbf{d}_j).$$

Define $$F_j(\mathbf{s}_{j-1}) = \max_{\mathbf{d}_N, \ldots, \mathbf{d}_j} (f_N \, o \, f_{N-1} \, o \ldots o \, f_j)$$

to postulate forward recursion formulae as

$$F_j(\mathbf{s}_{j-1}) = \max_{\mathbf{d}_j} [f_j\,(\mathbf{s}_{j-1}, \mathbf{d}_j) \; o \; F_{j+1}\,(\mathbf{s}_j)]\,,\; 1 \le j \le N-1$$

$$F_N\,(\mathbf{s}_{N-1}) = F_N\,(\mathbf{s}_{N-1}, \mathbf{d}_N).$$

Using these notations, the required optimum value of ψ_j is denoted by $F_1(\mathbf{s}_0)$ which can be obtained recursively through stage $j = N-1, \ldots, 2, 1$.

The forward recursion approach is explained by solving the numerical example which is solved by backward recursion approach earlier.

Q. 1. Describe the recursive equation approach to solve the dynamic programming problem. **[Raj. Univ. (M. Phil.) 92]**

2. State Bellman's principle of optimality. Explain the forward and backward recursion method. **[Meerut 2002; Delhi (OR) 93]**

3. What is dynamic programming relation ? Describe the general process of bactward recursion. **[IGNOU 2001 (June)]**

Example 20. *Minimize* $\mathbf{z} = y_1^2 + y_2^2 + y_3^2$ *subject to* $y_1 + y_2 + y_3 \ge 15\,;\, y_1, y_2, y_3 \ge 0$ *by forward recursion.* **[Kanpur 2000; Agra 97; I.A.S. (Main) 95; Raj. (M. Phil) 91]**

Solution. As usual, define state variables and put stage transformation as

$$s_1 = s_0 + y_1 \,,\ s_2 = s_1 + y_2 \,,\ s_3 = s_2 + y_3 \geq 15.$$

In this example, the forward recursion equation becomes

$$f_j(s_{j-1}) = \min_{y_j} [y_j^2 + F_{j+1}(s_j)] \,, j = 2, 1$$

with

$$F_3(s_2) = y_3^2 = (s_3 - s_2)^2.$$

Therefore, $F_2(s_1) = \min\limits_{y_2} [y_2^2 + F_3(s_2)] = \min\limits_{y_2} [y_2^2 + (s_3 - s_2)^2] = \min\limits_{y_2} [y_2^2 + (s_3 - s_1 - y_2)^2]$

Using calculus to find minima of the function of one variable :

$$F_2(s_1) = (s_3 - s_1)^2/2, \text{ for } y_2 = (s_3 - s_1)/2.$$

Also,

$$F_1(s_0) = \min_{y_1} [y_1^2 + \tfrac{1}{2}(s_3 - s_0 - y_1)^2]$$

Again, using calculus, $y_1 = (s_3 - s_0)/3$, and therefore, $F_1(s_0) = (s_3 - s_0)^2/3$.

Since $s_0 = 0$, $s_3 \geq 15$, $F_1(s_0)$ is minimum for $s_3 = 15$ or $y_1 = y_2 = y_3 = 5$.

Here it is observed that the final minimum is obtained as a function of input s_3 and the output s_0. Number of significant stage variables are now increased to four (instead of three). Thus, to solve the input problem by the forward recursion, number of state variables are $N + 1$ instead of N, s_0 is also appearing in the problem. However s_0 may be eliminated from the final result by considering that the result to be optimum with respect to s_N, and s_0 will adjust accordingly. Secondly, if the problem is of given out-put s_0 but solved through backward recursion, the input s_N will also get involved which can be removed from the final result. *Finally, when input and output are given as fixed, both forward and backward recursions are equally good.*

EXAMINATION PROBLEMS

1. Find the minimum value of
$z = y_1^2 + y_2^2 + \ldots + y_n^2$
subject to the constraints
$y_1 y_2 y_3 \ldots y_n = c$
and $y_j \geq 0$ for $j = 1, 2, \ldots, n$.
[Kanpur 96; Rohil. 94; Meerut 91]
[**Ans.** $(c^{2/n}, c^{2/n}, \ldots, c^{2/n})$ with $f_n(c) = nc^{2/n}$.]

2. Find the maximum value of
$z = x_1^2 + 2x_2^2 + 4x_3$
subject to the constraint
$x_1 + 2x_2 + x_3 \leq 8$,
$x_1, x, x_3 \geq 0$.
[Meerut (Maths) 99]
[**Ans.** (8, 0, 0) with $f_3^*(8) = 64$.]

3. Find the minimum value of
$x_1^2 + 2x_2^2 + 4x_3$
subject to the constraints;
$x_1 + x_2 + x_3 \geq 8$ and $x_1, x_2, x_3 \geq 0$.
[**Ans.** (2, 2, 2) with $f_3^*(8) = 20$.]

4. Find the maximum value of
$z = -x_1^2 - 2x_2^2 + 3x_2 + x_3$
subject to the conditions :
$x_1 + x_2 + x_3 \leq 1$ and $x_1, x_2, x_3 \geq 0$.
[**Ans.** (0, ½, ½) with $f_3^*(1) = 5/4$.]

5. Use method of dynamic programming to minimize $u_1^2 + u_2^2 + u_3^2$ subject to $u_1 + u_2 + u_3 \geq 10$, $u_1, u_2, u_3 \geq 0$. **[I.A.S. (Maths) 85]**
[**Hint.** See solved ***Example 20.***]

33.13. APPLICATIONS OF DYNAMIC PROGRAMMING

33.13-1 Application in Production

Dynamic programming approach can be effectively utilized in production systems. An example is given below.

Example 21. *Suppose there are n machines which can perform two jobs. If x of them do the first job, then they produce goods worth g(x) = 3x and if y of the machines perform the second job, then they produce goods worth h(y) = 2.5y. Machines are subject to depreciation, so that after performing the first job only a(x) = x/3 machines remain available and after performing the second job b(y) = ⅔ y machines remain available in the beginning of the second year. The process is repeated with remaining machines. Obtain the maximum total return after 3 years and also find the optimal policy in each year.* **[Delhi (OR) 92; Agra 93, 92]**

Solution. Here *first, second* and *third* year are considered as period 1, 2 and 3, respectively.

Let

$x_i =$ number of machines devoted to the job 1 in ith period.

$y_i =$ number of machines devoted to the job 2 in ith period.

$s_i =$ total number of machines in hand (available) at the beginning of ith period.

$f_n(s) =$ maximum possible return when there are n periods left with initial number of available machines being 's'.

The problem is now taken up through the use of backward recursion approach.

Step 1. Consider *third year* first, then s_3 is the number of machines available at the beginning of the year. Thus

$$f_1(s_3) = \max_{x_3, y_3} [3x_3 + 2.5y_3] \quad \text{...(33.41)}$$

$$\text{subject to } x_3 + y_3 \le s_3 \text{ and } x_3 \ge 0, y_3 \ge 0 \quad \text{...(33.42)}$$

It is obvious from linear programming that *extremal values of a linear function occur at corners of the constraint set.* Since the function $f_1(s_3) = 3x_3 + 2.5y_3$ is linear in x_3 and y_3, therefore maximum occurs at B $(s_3, 0)$ (see *Fig.* 33.7). Thus

$$f_1(s_3) = 3s_3 + 2.5 \times 0 = 3s_3 \quad \text{...(33.43)}$$

Hence, optimal decisions are :

$$x_3^* = s_3, \; y_2^* = 0 \text{ and } f_1(s_3) = 3s_3 \quad \text{...(33.44)}$$

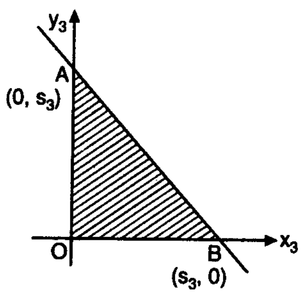

Fig. 33.7

Step 2. Now consider second year. Then number of machines available at the beginning of this period is s_2, and

$$f_2(s_2) = \max_{x_2, y_2} \left[3x_2 + 2.5y_2 + f_1\left(\frac{x_2}{3} + \frac{2y_2}{3}\right)\right]. \quad \text{...(33.45)}$$

(since x_2 and y_2 machines are utilised for two jobs, respectively; $x_2/3$ and $2y_2/3$ machines will remain available at the beginning of the next year).

Thus, by definition of f_1 as given in (33.43), we have

$$f_2(s_2) = \max_{x_2, y_2} \left[3x_2 + 2.5y_2 + 3\left(\frac{x_2}{3} + \frac{2y_2}{3}\right)\right] \quad \text{...(33.46a)}$$

or
$$f_2(s_2) = \max_{x_2, y_2} [4x_2 + 4.5y_2] \quad \text{...(33.46b)}$$

subject to the constraints :

$$x_2 + y_2 \le s_2 \text{ and } x_2 \ge 0, y_2 \ge 0 \quad \text{...(33.47)}$$

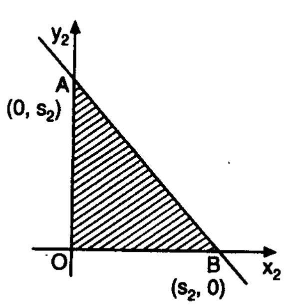

Fig. 33.8

Again, the objective function is linear and maximum occurs at the corner A $(0, s_3)$ (see *Fig.* 33.8). Thus

$$f_2(s_2) = 4.5\, s_2 \quad \text{...(33.48)}$$

and optimum decisions are :

$$x_2^* = 0 \text{ and } y_2^* = s_2 \quad \text{...(33.49)}$$

Step 3. Now in first year, the total number of available machines at the beginning of the period is s_1, and

$$f_3(s_1) = \max_{x_1, y_1} \left[3x_1 + 2.5y_1 + f_2\left(\frac{x_1}{3} + \frac{2y_1}{3}\right)\right]$$

$$= \max_{x_1, y_1} \left[3x_1 + 2.5y_1 + 4.5\left(\frac{x_1}{3} + \frac{2y_1}{3}\right)\right]$$

[using definition of f_2 as given in (33.48)]

$$= \max_{x_1, y_1} [4.5x_1 + 5.5y_1]$$

Hence,

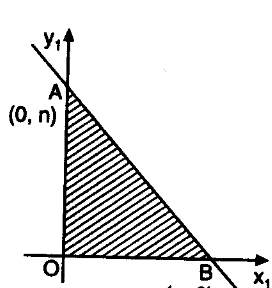

Fig. 33.9

$$f_3(s_1) = \max_{x_1, y_1} [4.5x_1 + 5.5y_1] \quad ...(33.50)$$

$$\text{subject to} \quad x_1 + y_1 \le s_1 \text{ and } x_1 \ge 0, y_1 \ge 0 \quad ...(33.51)$$

As explained earlier, objective function is linear. Thus, maximum occurs at the corner A (0, n) (see *Fig.* 33.9).

Therefore,

$$f_3(s_1) = f_3(n) = 5.5n \quad ...(33.52)$$

Thus, optimal decisions are

$$x_1^* = 0, y_1^* = n \quad .... (33.53)$$

and $s_2^* = \frac{2}{3} y_1^* = \frac{2}{3} n, x_2^* = 0$

Therefore, $y_2^* = s_2^* = \frac{2}{3} n$

Also, $s_3^* = \frac{2}{3} y_2^* = \frac{4}{9} n$

$x_3^* = s_3^* = \frac{4}{9} n, \; y_3^* = 0.$

Now, the complete solution of this problem is summarized and optimal policies for three periods are given in *Table* 33.3.

Table 33.3

Period 1	Period 2	Period 3
$x_1^* = 0$	$x_2^* = 0$	$x_3^* = 4n/9$
$y_1^* = n$	$y_2^* = 2n/3$	$y_3^*/0$

Maximum possible return $= f_3(n) = 5.5\ n$.

Q. Formulate a manpower loading problem as a dynamic programming problem.

33.13-2. Application in Inventory Control

Deterministic inventory models were considered in *Chapter* 2 (*Unit 4*) for constant demand of an item. If models are considered in which the demand is known exactly but different in each period, the solution of such models become somewhat more complicated. Such inventory models may be easily solved by using the dynamic programming technique. The procedure is explained by the following example :

Example 22. *A man is engaged in buying and selling identical items. He operates from a warehouse that can hold 500 items. Each month he can sell any quantity that he chooses up to the stock at the beginning of the month. Each month, he can buy as much as he wishes for delivery at the end of the month so long as his stock does not exceed 500 items. For the next four months, he has the following error-free forecasts of cost sales prices :*

Month :	i	1	2	3	4
Cost :	c_i	27	24	26	28
Sale Price :	p_i	28	25	25	27

If he currently has a stock of 200 units, what quantities should he sell and buy in next four months ? Find the solution using dynamic programming. **[Meerut (Maths.) 99, 96; Delhi (OR) 93, (M.B.A.) April 85; Rohilkhand 90]**

Solution. Here *first, second, third* and *fourth* month are denoted as period 1, 2, 3 and 4 respectively.

Let

x_i = amount to be sold during the month i — p_i = sale price in the month i

y_i = amount to be ordered during the month i — c_i = *purchase price in the* ith month, and

b_i = stock level in the beginning of month i — H = *warehouse capacity.*

Let $f_n(b_n)$ be the maximum possible return when there are n months to precede and initial stock is b_n.

The problem will be taken up as backward, *i.e.* consider i = 4 first and i = 1 last.

Thus $$f_1(b_n) = \max_{x_n, y_n} [p_n x_n - c_n y_n]$$

where $b_n \geq x_n$, $b_n - x_n + y_n \leq H$.

Also, $f_n(b_n) = \max_{x_n, y_n} [p_n x_n - c_n y_n + f_{n-1}(b_n - x_n + y_n)]$

For $n = 1$, $f_1(b_1) = \max_{x_1, y_1} [p_1 x_1 - c_1 y_1]$,

Obviously, $y_1 = 0, x_1 = b_1$.

Therefore, $f_1(b_1) = p_1 b_1 = 27b_1$, and $b_1 = b_2 - x_2 + y_2$

For $n = 2$, $f_2(b_2) = \max_{x_2, y_2} [p_2 x_2 - c_2 y_2 + f_1(b_2 - x_2 + y_2)]$

where $y_2 \leq H - b_2 + x_2 \leq 500 - b_2 + x_2$.

Therefore, $f_2(b_2) = \max_{x_2} [26b_2 - x_2 + 500] = 26b_2 + 500$ (taking $x_2 = 0$ for maximum)

and $b_2 = b_3 - x_3 + y_3$.

For $n = 3$,

$$f_3(b_3) = \max_{x_3, y_3} [p_3 x_3 - c_3 y_3 + f_2(b_3 - x_3 + y_3)] = \max_{x_3, y_3} [25x_3 - 24y_3 + 26(b_3 - x_3 + y_3) + 500]$$

$$= \max_{x_3, y_3} [26b_3 - x_3 + 2y_3 + 500], \text{where } y_3 \leq 500 - b_3 + x_3$$

$$= \max_{x_3} [26b_3 - x_3 + 2(500 - b_3 + x_3) + 500] \max_{x_3} [24b_3 + x_3 + 1500]$$

$$= 25b_3 + 1500 \quad (\text{since } b_3 \geq x_3, \text{ therefore } b_3 = x_3 \text{ for maximum})$$

But, $b_3 = b_4 - x_4 + y_4$. Now, taking $n = 4$,

$$f_4(b_4) = \max_{x_4, y_4} [p_4 x_4 - c_4 y_4 + f_3(b_4 - x_4 + y_4)]$$

$$= \max_{x_4, y_4} [28x_4 - 27y_4 + 25(b_4 - x_4 + y_4) + 1500] = \max_{x_4, y_4} [25b_4 + 3x_4 - 2y_4 + 1500]$$

$$= [25b_4 + 3b_4 + 1500] \quad (\text{since } y_4 = 0, x_4 = b_4 \text{ for maximum})$$

$$= 28b_4 + 1500.$$

It is given that

	$b_4 = 200$	$x_4 = 200, y_4 = 0.$
Therefore,	$b_3 = 200 - 200 + 0 = 0$	$x_3 = 0, y_3 = 500,$
	$b_2 = 0 - 0 + 500 = 500$	$x_2 = 0, y_2 = 0,$
	$b_1 = 500 - 0 - 0 = 500$	$x_1 = 500$, $y_1 = 0.$

Thus, the required solution is given in *Table* 33.4.

Table 33.4.

Month :	1	2	3	4
Purchase :	0	500	0	0
Sales :	700	0	500	500

Maximum possible return = 28 × 200 + 1500 = 7100. **Ans.**

Q. Discuss the dynamic programming approach to solve an inventory problem with illustration.

33.13-3. Application in Linear Programming **[Meerut 95]**

As discussed in **Unit 2,** the general linear programming problem is :

$$\textit{Maximize } \mathbf{z} = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \qquad ...(33.54)$$

subject to the constraints

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\le b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\le b_2 \\ \vdots \qquad \vdots \qquad \vdots \quad &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\le b_m \\ x_1 \ge 0, x_2 \ge 0, \ldots, x_n &\ge 0 \end{aligned}\right\} \qquad \ldots(33.55)$$

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

Let each activity, j (1, 2, ... , n) be a stage. The level of activity, x_j (≥ 0), represents decision variables (alternatives) at stage j. Since x_j is continuous, each stage possesses an infinite number of alternatives within the feasible region.

Since the linear programming problem is an allocation problem, states may be defined as the amounts of resources to be allocated to the *current stage* and *succeeding stages*. This will result in a backward functional (recursive) equation. Since there are m resources, stages must be represented by an m-dimensional vector.

Further, let $(\beta_{1j}, \beta_{2j}, \ldots, \beta_{mj})$ be the states of the system at stage j in accordance with the definition *i.e.*, amounts of resources 1, 2, 3, ..., m, respectively, are allocated to stage $j, j+1, \ldots, n$. Using the backward recursive equation, let $f_j(\beta_{1j}, \beta_{2j}, \ldots, \beta_{mj})$ be the optimum value of the objective function (33.54) for stages (activities) $j, j+1, \ldots, n$ for given states $\beta_{1j}, \beta_{2j}, \ldots, \beta_{mj}$.

Thus,

$$f_n(\beta_{1n}, \beta_{2n}, \ldots, \beta_{mn}) = \max_{0 \le a_{ij} x_n \le \beta_{in}} [c_n x_n], \; i = 1, 2, \ldots, m \qquad \ldots(33.56)$$

$$f_j(\beta_{1j}, \beta_{2j}, \ldots, \beta_{mj}) = \max_{\substack{0 \le a_{ij} x_j \le \beta_{ij} \\ (i=1,2,\ldots,m)}} [c_j x_j + f_{j+1}(\beta_{1j} - a_{1j}x_j, \ldots, \beta_{mj} - a_{mj}x_j)] \qquad \ldots(33.57)$$

for $j = 1, 2, 3, \ldots, n-1$, where it is understood that $0 \le \beta_{ij} \le b_i$ for all i and j.

Thus, a recursive equation (33.57) is obtained and can be used to solve the linear programming problem by the dynamic programming approach.

Example 23. *Solve the following linear programming problem by dynamic programming approach.*

Maximize $z = 2x_1 + 5x_2$, *subject to the constraints* $2x_1 + x_2 \le 43$, $2x_2 \le 46$ and $x_1 \ge 0, x_2 \ge 0$

[JNTU (Mech. & Prod.) 2004; Meerut 98]

Solution. Since there are two resources, the states of the equivalent dynamic programming problem can be described by two variables only.

Let (β_1, β_2) describe the states $j (= 1, 2)$.

Thus, for $j = 2$, we have

$$f_2(\beta_{12}, \beta_{22}) = \max_{\substack{0 \le x_2 \le \beta_{12} \\ 0 \le 2x_2 \le \beta_{22}}} [5x_2] \qquad \ldots(33.58)$$

Since $\quad x_2 \le \min[\beta_{12}, \beta_{22}/2]$ and $f_2[\beta_{12} - x_2, \beta_{22}] = 5x_2$,

then $\quad f_2(\beta_{12}, \beta_{22}) = \max_{x_2} f_2[\beta_{12} - x_2, \beta_{22}] = 5 \min[\beta_{12}, \beta_{22}/2] \qquad \ldots(33.59)$

and $\quad x_2^* = \min[\beta_{12}, \beta_{22}/2] \qquad \ldots(33.60)$

Now

$$f_1(\beta_{11}, \beta_{21}) = \max_{\substack{0 \le 2x_1 \le \beta_{11} \\ 0 \le 0x_1 \le \beta_{21}}} [2x_1 + f_2(\beta_{11} - 2x_1, \beta_{21} - 0)]$$

$$= \max_{0 \le 2x_1 \le \beta_{11}} [2x_1 + 5 \min(\beta_{11} - 2x_1, \beta_{21}/0)] \quad \text{[by def. of } f_2 \text{ from (33.59)]}$$

Since this is the last stage, then $\beta_{11} = 43$, $\beta_{21} = 46$.

Thus $\quad x_1 \le \beta_{11}/2 = 21.5$ and $f_1(\beta_{11} - 2x_1, \beta_{21}/2) = f_1(43 - 2x_1, 46/2)$

$$= 2x_1 + 5 \min(43 - 2x_1, 46/2)$$

$$= 2x_1 + \begin{cases} 5 \times 23, & 0 \le x_1 \le 10 \\ 5(43 - 2x_1), & 10 \le x_1 \le 21.5 \end{cases}$$

$$= \begin{cases} 2x_1 + 11{\cdot}5, 0 \le x_1 \le 10 \\ -8x_1 + 21{\cdot}5, 10 \le x_1 \le 21{\cdot}5 \end{cases} \qquad \begin{pmatrix} \text{(i) } 43 - 2x_1 \ge 23 \Rightarrow 0 \le x_1 \le 10 \\ \text{(ii) } 43 - 2x_1 \le 23 \Rightarrow x_1 \ge 10 \end{pmatrix}$$

Hence for given range of x_1,

$$f(\beta_{11}, \beta_{21}) = f(43, 46)$$
$$= \max_{x_1} (2x_1 + 115, -8x_1 + 215)$$
$$= \max_{x_1 = 10} [2(10) + 115, -8(10) + 215] = 135 \quad (at\ x_1^* = 10)$$

To obtain x_2^*, we observe that

$$\beta_{12} = \beta_{11} - 2x_1 = 43 - 20 = 23, \quad \beta_{22} = \beta_{21} - 0 = 46$$

and $$x_2^* = \min[\beta_{12}, \beta_{22}/2] = \min[23, 46/2] = 23.$$

Thus optimal solution is given by $z^* = 135$, $x_1^* = 10$, $x_2^* = 23$.

Alternative. Since there are two resources, the states of equivalent dynamic programming problem can be described by two variables only.

Let (u_j, v_j) describe state j (= 1, 2). Thus, $f_2(u_2, v_2) = \max\limits_{\substack{0 \le x_1 \le u_2 \\ 0 \le 2x_2 \le v_2}} [5x_2]$

Since $x_2 \le \min(u_2, v_2/2)$ and $f_2(x_2 \mid \text{given } u_2, v_2)$ then

$$f_2(u_2, v_2) = \max_{x_2} f_2(x_2 \mid \text{given } u_2, v_2) = 5 \min(u_2, v_2/2) \qquad ...(33.59)$$

and $$x_2^* = \min(u_2, v_2/2) \qquad ...(33.60)$$

Now, $$f_1(u_1, v_1) = \max_{\substack{0 \le 2x_1 \le u_1 \\ 0 \le 0x_1 \le v_1}} [2x_1 + f_2(u_1 - 2x_1, v_1 - 0)]$$

$$= \max_{0 \le 2x_1 \le u_1} [2x_1 + 5 \min(u_1 - 2x_1, v_1/2)] \qquad \{\text{by definition of } f_2 \text{ from (33.59)}\}$$

Since this is the last stage, then $u_1 = 43$, $v_1 = 46$.

Thus, $$x_1 \le \tfrac{1}{2}u_1 = 21.5,$$

and $$f_1(x_1 \mid \text{given } u_1, v_1) = f_1(x_1 \mid \text{given } u_1 = 43, v_1 = 46)$$
$$= 2x_1 + 5 \min(43 - 2x_1, 46/2)$$
$$= 2x_1 + \begin{cases} 5(23), \text{ for } 0 \le x_1 \le 10 \\ 5(43 - 2x_1), \text{ for } 10 \le x_1 \le 21.5 \end{cases}$$
$$= \begin{cases} 2x_1 + 115, \ 0 \le x_1 \le 10 \\ -8x_1 + 215, \ 10 \le x_1 \le 21.5 \end{cases}$$

Hence for given range of x_1,

$\therefore$ $$f(u_1, v_1) = f_1(43, 46) = \max_{x_1} (2x_1 + 115, -8x_1 + 215)$$
$$= \max[2(10) + 115, -8(10) + 215] = 135$$

which is achieved at $x_1^* = 10$.

To obtain x_2^*, it is observed that $u_2 = u_1 - 2x_1 = 43 - 20 = 23$, $v_2 = v_1 - 0 = 46$

and $$x_2^* = \min(u_2, v_2/2) = \min(23, 46/2) = 23.$$

Thus, the optimal solution is given by $\mathbf{z}^* = 135$, $x_1 = 10$, $x_2 = 23$.

This example shows that in comparison to *simplex method* it is too much difficult to solve a linear programming problem by the *dynamic programming* approach.

Example 24. *Use dynamic programming to solve the L.P.P. :*

Max $z = x_1 + 9x_2$, *subject to the constraints :* $2x_1 + x_2 \le 25, x_2 \le 11 ; x_1, x_2 \ge 0.$

[Meerut 96 BP, 93 P; Agra 95; Rewa (M.P.) 93]

Solution. The problem has two resources and two decision variables. The states of the equivalent dynamic programming are β_{1j} and β_{2j} for $j = 1, 2$. Thus

$$f_2(\beta_{12}, \beta_{22}) = \max \{9x_2\}$$

where maximum is taken over $0 \le x_2 \le 25$ and $0 \le x_2 \le 11$. That is,

$$f_2(\beta_{12}, \beta_{22}) = 9 \max \{x_2\} = 9 \max \{25, 11\}$$

Since the maximum of x_2 satisfying the conditions of $x_2 \le 25$ and $x_2 \le 11$ is the minimum of {25 and 11}.

Therefore, $x_2^* = 11$.

Now, $$f_1(\beta_{11}, \beta_{21}) = \max \{x_1 + f_2(\beta_{11} - 2x_1, \beta_{21} - 0)\}$$

where maximum is taken over $0 \le x_1 \le 25/2$.

At this last stage, substitute the value of $\beta_{11} = 25$ and $\beta_{21} = 11$. Therefore,

$$f_1(25, 11) = \max \{x_1 + 9 \min (25 - 2x_1, 11)\}$$

Now, $$\min (25 - 2x_1, 11) = \begin{cases} 11, & \text{for } 0 \le x_1 \le 7 \\ 25 - 2x_1, & \text{for } 7 \le x_1 \le 25/2 \end{cases}$$

Therefore, $$x_1 + 9 \min (25 - 2x_1, 11) = \begin{cases} x_1 + 99, & \text{for } 0 \le x_1 \le 7 \\ 225 - 17x_1, & \text{for } 7 \le x_1 \le 25/2 \end{cases}$$

Since the maximum of both $x_1 + 99$ and $225 - 17x_1$ occurs at $x_1 = 7$, therefore

$$f_1(25, 11) = 7 + 9 \min (11, 11) = 106, \text{ at } x_1^* = 7$$

$$x_2^* = \min (25 - 2x_1^*, 11) = \min (11, 11) = 11$$

Hence the optimum solution is $x_1^* = 7, x_2^* = 11$ and max $z = 106$.

Q. Explain the concept of dynamic programming and the relation between 'dynamic' and 'linear' programming problems. Show how to solve a linear programming problem by dynamic programming technique. **[I.A.S. (Main) 79]**

EXAMINATION PROBLEMS

Solve the following linear programming problems by dynamic programming:

1. Max $z = 8x_1 + 7x_2$, subject to the constraints :
   $2x_1 + x_2 \le 8$, $5x_1 + 2x_2 \le 15$ and $x_1, x_2 \ge 0$
   [Ans. $x_1^* = 0$, $x_2^* = 7.5$ and max $z = 52.5$**]** **[JNTU (MCA III) 2004, (B. Tech.) 2003]**
2. Max $z = 3x_1 + 5x_2$, subject to the constraints : $x_1 \le 4$; $x_2 \le 6$; $3x_1 + 2x_2 \le 18$; $x_1, x_2 \ge 0$.
   [Agra 99, 98; Rohilkhand 93]
   [Hint. *The problem consists of three resources and two decision variables. The states of the equivalent dynamic programming problem are* $(\beta_{1j}, \beta_{2j}, \beta_{3j})$ for $j = 1, 2$.]
   [Ans. $x_1^* = 2$, $x_2^* = 6$ and max $z = 36$**]**
3. Max $z = 50x_1 + 100x_2$, subject to the constraints :
   $10x_1 + 5x_2 \le 2500$, $4x_1 + 10x_2 \le 2000$
   $x_1 + 3/2\, x_2 \le 450$, and $x_1, x_2 \ge 0$.
   [Ans. $x_1^* = 187.5$, $x_2^* = 125.0$ and max $z = 21875$**]**
4. Max $z = 3x_1 + x_2$, subject to the constraints :
   $2x_1 + x_2 \le 6$; $x_1 \le 2$; $x_2 \le 4$ and $x_1, x_2 \ge 0$.
   [Ans. $x_1 = x_2 = 2$ and max $z = 8$**]**
   [Meerut 97P, 94; Delhi (OR) 93]
5. Max $z = 3x_1 + 7x_2$ subject to the constraints :
   $x_1 + 4x_2 \le 8$, $x_2 \le 2$ and $x_1, x_2 \ge 0$.
   [Ans. $x_1 = 8$, $x_2 = 0$; max $z = 24$**]**
6. Max $z = 2x_1 + 5x_2$ subject to the constraints :
   $3x_1 + x_2 \le 2$, $x_2 \le 3$ and $x_1, x_2 \ge 0$.
   [Ans $x_1 = 3$, $x_2 = 3$; max $z = 21$**]**
7. Solve the following linear programming problem by applying dynamic programing procedures. Explain the assumptions you make :
   Max $z = 50x_1 + 100x_2$, subject to the constraints : $2x_1 + 3x_2 \le 48$, $x_1 + 3x_2 \le 42$, $x_1 + x_2 \le 21$ and $x_1, x_2 \ge 0$.
   [Ans. $x_1 = 6$, $x_2 = 12$; max $z = 60$**]**

8. Solve the following linear programming problem by dynamic programming technique :
Max. $5x_1 + 3x_2$, subject to $x_1 \le 12$, $x_2 \le 8$, $2x_1 + 3x_2 \le 36$; $x_1, x_2 \ge 0$.

9. Use dynamic programming to solve :
Min. $3x_1 + 5x_2$, subject to $-3x_1 + 4x_2 \le 12$, $-2x_1 + x_2 \le 2$, $2x_1 + 3x_2 \ge 12$, $0 \le x_1 \le 4$, $x_2 \ge 2$.
[Meerut 98 BP]

33.13-4. Application in Reliability

Following example is presented in order to demonstrate the application of *dyanamic programming* in *reliability*.

Example 25. ***(Reliability Problem)*** *Consider the problem of designing an electronic device consisting of three main components. The three components are arranged in series so that the failure of one of the components will result in the failure of the whole device. Therefore, it is decided that the reliability (prob. of no failure) of the device can be improved by installing parallel (stand-by) units on each component. Each component may be installed at most 3 parallel units. The total capital (in thousand Rs) available for the device is 10. Following data is available :*

m_i	$i=1$		$i=2$		$i=3$	
	r_1	c_1	r_2	c_2	r_3	c_3
1	.5	2	.7	3	.6	1
2	.7	4	.8	5	.8	2
3	.9	5	.9	6	.9	3

Here m_i is the number of parallel units placed with the ith component, r_i is the reliability of the component and c_i is the cost for the ith component. Determine m_i which will maximize the total reliability of the system without exceeding the given capital.

Solution. ***Step 1. Formulation of the problem :***

Let R be the total reliability of a system of n components arranged in series and m_i parallel units per component i ($i = 1, 2, 3$). Thus the problem is :

Maximize $R = r_1 r_2 r_3$, *subject to the constraint* $c_1 + c_2 + c_3 \le c$,

where c is the total capital available.

Step 2. To obtain Recursive Relationships :

Let $x_i \rightarrow$ defines the capital allocated to stages 1, 2, 3, ... , i.

$r_i(c_i) \rightarrow$reliability r_i is a function of c_i.

$f_i(x_i) \rightarrow$reliability of components (stages) 1 through i inclusive, given that $0 \le x_i \le c$.

Since x_i are the states of the system, the recursive equations are then given by

$$f_1(x_1) = \max_{\substack{m_1 \\ 0 \le c_1 \le x_1}} \{r_1(c_1)\} \text{ for } m_1 = 1, 2, 3$$

and

$$f_i(x_i) = \max_{\substack{m_i \\ 0 \le c_i \le x_i}} \{r_i(c_i) . f_{i-1}(x_i - c_i)\} \; i = 2, 3, \ldots, \text{ and } m_i = 1, 2, 3.$$

Since m_i and c_i both are given in discrete units, the tabular computations are performed :

Stage 1

	$f_1(x_1) = r_1(c_1)$			Maximum reliability	
s / x_1	$m_1 = 1$ $r_1 = .5, c_1 = 2$	$m_1 = 2$ $r_1 = .7, c_1 = 4$	$m_1 = 3$ $r_1 = .9, c_1 = 5$	$f_1(x_1)$	m_1^*
0	—	—	—	—	—
1	—	—	—	—	—
2	0.5	—	—	0.5	1
3	0.5	—	—	0.5	1
4	0.5	0.7	—	0.7	2
5	0.5	0.7	0.9	0.9	3
6	0.5	0.7	0.9	0.9	3
7	0.5	0.7	0.9	0.9	3
8	0.5	0.7	0.9	0.9	3
9	0.5	0.7	0.9	0.9	3
10	0.5	0.7	0.9	0.9	3

Stage 2

x_2 \ s	$f_2(x_2) = r_2(c_2) f_1(x_2 - c_2)$ $m_2 = 1$, $r_2 = 0.7, c_2 = 3$	$m_2 = 2$, $r_2 = 0.8, c_2 = 5$	$m_2 = 3$, $r_2 = 0.9, c_2 = 6$	Maximum reliability $f_2(x_2)$	m_2^*
0	—	—	—	—	—
1	—	—	—	—	—
2	—	—	—	—	—
3	.7 × (–) = —	—	—	—	—
4	.7 × (–) = —	—	—	—	—
5	.7 × .5 = .35	.8 × (–) = —	—	.35	1
6	.7 × .5 = .35	.8 × (–) = —	.9 × (–) = —	.35	1
7	.7 × .7 = .49	.8 × .5 = .40	.9 × (–) = —	.49	1
8	.7 × .9 = .63	.8 × .5 = .40	.9 × .5 = .45	.63	1
9	.7 × .9 = .63	.8 × .7 = .56	.9 × .5 = .45	.63	1
10	.7 × .9 = .63	.8 × .9 = .72	.9 × .7 = .63	.72	2

Stage 3

x_3 \ s	$f_3(x_3) = r_3(c_3) f_2(x_3 - c_3)$ $m_3 = 1$, $r_3 = 0.6, c_3 = 1$	$m_3 = 2$, $r_3 = 0.8, c_3 = 2$	$m_3 = 3$, $r_3 = 0.9, c_3 = 3$	Maximum reliability $f_3(x_3)$	m_3^*
0	—	—	—	—	—
1	.6 × (–) = –	—	—	—	—
2	.6 × .(–) = –	.8 × .(–) = .–	—	—	—
3	.6 × .(–) = –	.8 × .(–) = .–	.9 × .(–) = .–	—	—
4	.6 × .(–) = –	.8 × .(–) = .–	.9 × .(–) = .–	—	—
5	.6 × .(–) = –	.8 × .(–) = .–	.9 × .(–) = .–	—	—
6	.6 × .35 = .210	.8 × .(–) = .–	.9 × .(–) = .–	.210	1
7	.6 × .35 = .210	.8 × .35 = .280	.9 × .(–) = .–	.280	2
8	.6 × .49 = .294	.8 × .35 = .280	.9 × .35 = .315	.315	3
9	.6 × .63 = .378	.8 × .49 = .392	.9 × .35 = .315	.392	2
10	.6 × .63 = .378	.8 × .63 = .504	.9 × .49 = .441	.504	2

The optimal solution is therefore given by $m_1^* = 3$, $m_2^* = 1$ and $m_3^* = 2$ with the maximum reliability 0.504.

33.13-5. Application in Continuous Systems

The dynamic programming approach can be applied to infinitely multistage systems also. The N-discrete stages of a system may differ infinitesimally from each other, and stages may vary continuously as $N \to \infty$. A model of continuous infinitely multistage process is analogous to *forward recursion approach* applicable to discrete models as discussed earlier. A continuous model can be obtained through the passage of the discrete to the continuous as given below.

Analogy between Discrete and Continuous Systems

	systems	*Discrete*	*Continuous*
1.	Stage	index $j = 0, 1, \ldots, N$	parameter $t_0 \le t \le t_1$
2.	Decision variable	d_j	$\mathbf{d}(t)$
3.	State variable	$\mathbf{s}_j$	$\mathbf{s}(t)$
4.	Stage return	$f_j(\mathbf{s}_{j-1}, \mathbf{d}_j)$	$\int_t^{t+\Delta t} f(t, \mathbf{s}, \mathbf{d})\, dt$
5.	Total return	$\sum_{j=1}^{N} f_j$	$\int_{t_0}^{t_1} f(t, \mathbf{s}, \mathbf{d})\, dt$
6.	Stage transformation.	$\mathbf{s}_j = T_j(\mathbf{s}_{j-1}, \mathbf{d}_j)$	$ds/dt = G(t, \mathbf{s}, \mathbf{d})$

Thus, the continuous system is of the form :

$$\text{Maximize } \mathbf{z} = \int_{t_0}^{t_1} f(t, \mathbf{s}, \mathbf{d})\, dt, \text{ subject to the condition}$$

$$\frac{ds}{dt} = G(t, \mathbf{s}, \mathbf{d}),\ t_0 \le t \le t_1$$

where **s** is an n-vector in E^n, **d** is an m-vector e^m, and G denotes n functions $g_1, g_2, \dots, g_n$.

With the prescribed value $\mathbf{s}(t_0) = \mathbf{s}_0$, this is a typical problem of optimal control and calculus of variations.

Thus, there is a close relationship between *dynamic programming, calculus of variations*, and *optimal control.*

The forward recursion formula of dynamic programming for the continuous form is obtained below :

If the notation of the descrete problem :

$$F_j(\mathbf{s}_{j-1}) = \max_{\mathbf{d}_N, \dots, \mathbf{d}_j} (f_N \, o \, f_{N-1}, \dots, o \, f_j)$$

is replaced by its continuous analogue

$$\mathbf{z}(t, \mathbf{s}) = \max_{\mathbf{d}} \int_{t_0}^{t_1} f(t, \mathbf{s}, \mathbf{d})\, dt,$$

then the discrete recursion formula

$$F_j(\mathbf{s}_{j-1}) = \max_{\mathbf{d}_j} [f_j(\mathbf{s}_{j-1}, \mathbf{d}_j) \, o \, F_{j+1}(\mathbf{s}_j)]$$

can be replaced by

$$\mathbf{z}(t, \mathbf{s}) = \max_{\mathbf{d}} \left[\int_{t}^{t+\Delta t} f(t, \mathbf{s}, \mathbf{d})\, dt + \mathbf{z}(t + \Delta t, \mathbf{s} + \Delta \mathbf{s}) \right]$$

These are fundamental equations occurring in the theory of optimal control.

33.14. CHARACTERISTICS OF DYNAMIC PROGRAMMING PROBLEM

The characteristics of dyanamic programming problem may be outlined as follows :

[JNTU (MCA III) 2004; Delhi (Stat.) 96; (OR) 93]

1. The problem can be divided into stages, with a policy decision required at each stage.
2. Each *stage* has a number of *states* associated with it.
3. The effect of the policy decision at each stage is to transform the current state into a state associated with the next stage.
4. Given the current stage, an optimal policy for remaining stages is independent of the policy adopted in the previous stages.
5. The solution procedure begins by finding an optimal policy for each state of the last stage.
6. A functional equation is available which identifies the optimal policy for each state with n stages remaining, given the optimal policy for each state with $(n-1)$ stages left.
7. Using this functional equation, the solution procedure moves backward stage-by-stage, each time finding the policy when starting at the initial stage.

SELF-EXAMINATION PROBLEMS

1. Use dynamic programming to find the value of

   $\max \mathbf{z} = y_1 y_2 y_3$, subject to the constraints : $y_1 + y_2 + y_3 = 5$, and $y_1, y_2, y_3 \ge 0$.

   [JNTU (B. Tech.) 2003 (Type)]

   [**Hint.** $f_1(x_1) = \max_{z_1 = x_1} (z_1)$ and $f_j(x_j) = \max_{0 \le z_j \le x_j} \{z_j f_{j-1}(x_j - z_j)\}$, $j = 1, 2, 3$

   [**Ans.** (5/3, 5/3, 5/3) with $f_3(5) = (5/3)^3$]

2. Formulate the following problem as a dynamic programming problem.

   Minimize $\mathbf{z} = (x_1 + 2)^2 + x_2 x_3 - (x_4 - 5)^2$ subject to $x_1 + x_2 + x_3 + x_4 \le 5$, and x_1, x_2, x_3, x_4 are non-negative integers.

   Find the optimum solution. What is the optimum solution if the right hand side of the constraint is 3 instead of 5 ?

3. Solve the following problem by dynamic programming :

$$\max \sum_{n=1}^{4}(4d_n - nd_n^2), \text{ subject to the constraints } \sum_{n=1}^{4} d_n = 10,\ d_n \geq 0.$$

[**Ans.** 8]

4. Illustrate the dynamic programming appraoch by solving the following problem

$$\max\ 12x_1^3 + 27x_2^3 + 14x_3^3$$

in non-negative x_i such that $\Sigma x_i = 1$, $i = 1, 2, 3$.

5. (i) Develop the functional equation to determine $m_i\ (\geq 0)$ so as to maximize

$$z = \sum_{i=1}^{n} m_i\,(p_i/m_n)^{\alpha} \text{ subject to the constraints : } m_1 + m_2 + \ldots + m_n = M.$$

(ii) Develop the functional equation to determine $m_1, m_2, \ldots, m_n$ so that

$$\frac{\Sigma m_i\,(p_i)^2}{m_n} \text{ is maximum such that } m_1, m_2, \ldots, m_n = M.$$

[**Ans.** (i) $m_1 = 0, m_2 = 0, \ldots, m_{n-1} = 0, m_n = M$, and $f_n(M) = M\,(p_n/m_n)^{\alpha}$] **(Raj. Univ. (M. Phil.) 93)**

6. Solve the following linear programming problems by dynamic programming

(i) Max $z = 2x_1 + 3x_2$, subject to the constraints :
$x_1 - x_2 \leq 1,\ x_1 + x_2 \leq 3,\ x_1, x_2 \geq 0$
and $x_1, x_2, x_3 \geq 0$
[**Ans.** $x_1 = 0, x_2 = 3$, max $z = 9$.]

(ii) Min $z = x_1 + 3x_2 + 4x_3$, subject to
$2x_1 + 4x_2 + 3x_3 \geq 60,\ 3x_1 + 2x_2 + x_3 \geq 60,$
$2x_1 + x_2 + 3x_3 \geq 90$, and $x_1, x_2\ x_3 \geq 0$

7. The total volume available in an aircraft for 3 types of item is 13 ft^3. The unit volume of item *A* is 2 ft.^{3}, that of item *B* is 3 ft.^{3} and that of item *C* is 2 ft^3. The cost of having a demand that occur when the system is out of stock is Rs. 600 for item *A*, Rs. 1200 for item *B*, and Rs. 800 for item *C*. The demand for each item is Poisson distribution with mean being 5, 2 and 2 for items *A, B and C* respectively. How many of each item should be loaded in order to minimize the expected stockout costs ?

8. A naval base is interested in stocking three important spare parts of submarines. These parts have volumes of 0, 1, 2 and 2 cubic feet respectively. A total of 10 cubic feet of storage space is available. The shortage costs are Rs. 800, Rs. 600 and Rs. 1300 respectively. The demand for each spare part has a Poisson distribution with means 4, 2 and 1 respectively. Determine how many of each spare part should be stocked so as to minimize the repeated shortage cost.

9. Maximize hydro-electric power $P(s)$, $s = (s_1, s_2, s_3)$, produced by building dams on three different river basins, where

$$P(s) = f_1(s_1) + f_2(s_2) + f_3(s_3),$$

and $f_i(s_i)$ is the power generated from the ith basin by investing Rs. s_i. The total budgetary provision is $s_1 + s_2 + s_3 \leq 3$. The functions f_1, f_2, f_3 are given in the following table. Integer solution of the problem is required.
[**Ans.** 8, 0, 2, 1]

s_i	0	1	2	3
f_1	0	2	4	6
f_2	0	1	5	6
f_3	0	3	5	6

10. Given four items $j\ (= 1, 2, 3, 4)$ with weights w_j per unit and values v_j per unit. Find the positive integer quantity of each item to be placed in a bag so that the total weight of the items does not exceed W and the total value is maximum. Take the following numerical data :
[**Ans.** 32, 2, 1, 1]

j	:	1	2	3	4	
w_j	:	1	3	4	6	$W = 19$
v	:	1	5	7	11	

11. We have a bomber and two enemy targets. A raid on a target '*A*' will result either in a fraction r_1 of the enemy's resources in *A* being destroyed or the bomber being shot down (before inflicting damage); the probability of the bomber surviving a mission to *A* being p_1. Target *B* is similarly associated with a fraction r_2 and a probability p_2. The enemy's resources initially are x at *A* and y at *B*. Find a functional equation to determine the optimal policy when the number of raids is limited to *N* and when the number can be inifinite.

12. Explain in brief the dynamic programming appraoch and pose the following problem on a dynamic programming problem approach and solve.

A dealer places an order with his wholesaler on the first of each month and obtains delivery one month later. The cost of holding inventory is c_1 per unit per month and the cost of shortage is c_2 per unit per month, shortages being carried over from one month to the next. If the monthly demand x is a random variable with density function $p(x)$, find the policy that minimises the long-term average costs per month.

13. (i) A man is engaged in buying and selling identical items, each of which requires considerable storage space. The buying and selling prices are indicated in the table below. He operates from a warehouse which has a capacity of 500 items. He can order on the 15th of each month, for delivery on the first day of the following month. During a month, he can also sell any amount upto his total stock on hand.

	January	February	March
Cost Prices (Rs.)	150	155	165
Sales Prices (Rs.)	165	165	185

If he starts the year with 200 items in stock, how much should he plan to purchase and sell each month, in order to maximize his profit for the first quarter of the year ?

(ii) Solve the above problem for the marked prices given below :

		Jan (15)	Feb (15)	March (15)	April (15)	May (15)	June (15)
Cost Prices (Rs.)	:	155	150	155	155	150	150
Sales Prices (Rs.)	:	155	155	160	170	175	170

Maximize his profits for the half year.

14. An item have five months selling period with the probability distribution (given below) of selling in each month.

Probability distribution of selling price in each month.

	Month				
Price	1	2	3	4	5
4	.10	.05	.05	.05	.05
5	.10	.10	.15	.05	.25
6	.20	.15	.30	.35	.30
7	.30	.25	.15	.25	.20
8	.15	.20	.15	.15	.15
9	.10	.15	.10	.10	.05
10	.05	.10	.10	.05	.00

(a) Calculate the expected price for each.

(b) Faced with these probability distributions for the price over the demand season, use a method of dynamic programming to determine an optimal selling policy.

15. The ABC corporation has *nine* salesmen who presently sell in *three* separate sales areas of Northern India. The profitability for each salesman in the *three* sales areas as as follows :

		No. of Salesman									
Area	1	0	1	2	3	4	5	6	7	8	9
	2	9	8	7	6	5	4	3	2	1	0
	3	0	1	2	3	4	5	6	7	8	9

		Profitability (in thousand of Rs.)									
Area	1	20	32	47	57	66	71	82	90	100	110
	2	135	125	115	104	93	82	71	60	50	40
	3	50	61	72	84	97	109	120	131	140	150

Determine the optimum allocation of salesmen in order to maximize profits.

16. The work load for the *local Job shop* is subject to considerable seasonal fluctuation. However, machine operators are difficult to hire and costly to train, so the manager is reluctant to lay-off workers during the slack seasons. He is likewise reluctant to maintain his peak season pay roll when it is not required. Furthermore, he is definitly opposed to overtime work on a regular basis. Since all work is done to custom orders if not possible to build up inventories during slack seasons. Therefore, the manager is in a dilemma as to what his policy should be regarding employment levels.

The following estimates are given for the man-power requirements during the four seasons of the year for the foreseeable future :

Season :	Spring	Summer	Autumn	Winter	Spring
Requirement :	255	220	240	200	255

Employment will not be permitted to fall below these levels. Any employment above these levels is wasted at an approximated cost of Rs. 2,000 per man per season. It is also estimated that the hiring and firing costs are such that the total cost of changing the level of employment from one season to the next is Rs. 200 times the square of the difference in employment levels. Fractional levels of employment are possible because of a few part time employees, and the above cost data also apply on a fractional basis.

The manager needs to determine what employment level should be in each season to minimize total cost.

17. In an *N*-stage rocket the weight *W* of the *j*th state is a function of the velocity increase v_j that takes place during the firing of that stage and the weight w_j of the burnt out $(j-1)$ stages, *i.e.* $[W_j, (v_j, w_j)]$. The final velocity to be attained after the last burning stage is *V* and the total weight of the rocket is to be minimum. Identify the space and decision variables and the stage transformation function in the problem. Obtain the recursion equation for optimization and show that minimum weight is given by

$$f_N(V) = \min_{v_N} [W_N \{v_N, f_{N-1}(V - v_N)\} + f_{N-1}(V - v_N)]$$

18. We have a machine that deteriorates with age and so we have to decide about the replacement policy. We have to own such a machine during each of the next 5 years. The operating cost $c(i)$ of a machine *i* years old at the beginning of the year; trade in value $t(i)$ received when such a machine is traded for a new machine at the start of the year and $s(i)$, the salvage value received for a machine that have just turned age *i* at the end of 5 years are given below :

i	0	1	2	3	4	5	6
$c(i)$	10	13	20	40	70	100	100
$t(i)$	—	32	21	11	5	0	0
$s(i)$	—	25	17	8	0	0	0

If a new machine costs 50 and we have now a machine which is two years old; what is the optimum policy of replacement ? Solve the problem by *Dynamic Programming.*

19. An enterprising young researcher believes that he has developed a system for winning a popular *Las Vegas* game. His colleagues do not believe that this is possible, so they have made a large bet with him. They bet that starting with three chips, he will not have at least five chips after three plays of the game. Each play of the game involves batting any desired number of available chips and then either winning or losing this number of chips. He believes that his system will give him a probability of 2/3 of winning a given play of the game. Find his optimal policy regarding how many chips to bet, if any, at each of the three plays of the game in order to maximize the probability of winning his bet with his colleagues.

20. 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 need to determine how many teams (if any) to allocate to each of these countries to maximize the total effectiveness of the five items. 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 items.

No. of Medical Teams	*Thousands of Additioanl Man-years of Life*		
	Country 1	*Country* 2	*Country* 3
0	0	0	0
1	45	20	50
2	70	45	70
3	90	75	80
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.

[Ans. *One* team to country 1, three taeams to country 2, and *one* team to country 3; and maximum effectiveness is 170]

21. An investor has Rs. 6000 to invest. This amount can be invested in any of three ventures available to him. But, he must invest in units of Rs. 1000. The potential return from investment in any one venture depends upon the amount invested according to the following table (all figures in thousands)

Amount Invested	*Return from Venture*		
	A	B	C
0	0	0	0
1	0.5	1.5	1.2
2	1.0	2.0	2.4
3	3.0	2.2	2.5
4	3.1	2.3	2.6
5	3.2	2.4	2.7
6	3.3	2.5	2.8

The investor wishes to imvest Rs. 6000 so that the return from investment is maximum. Formulate the above problem as a dynamic programming problem and find the optimum investment policy.

[Ans. Rs. 3000 in *A*, Rs. 1000 in *B* and Rs. 2000 in *C*; Max. return Rs. 6900].

22. Using dynamic programming approach, solve the reliability problem with the following data : The total capital available is 10 (in units of thousand rupees)

[Ans. Optimal solution : $M_1 = 3$, $M_2 = 1$ and $M_3 = 2$ with maximum reliability as 0.576]

M_i	$i = 1$		$i = 2$		$i = 3$	
	R	C	R	C	R	C
1	.6	2	.8	3	.7	1
2	.7	4	.8	5	.8	2
3	.9	5	.9	6	.9	3