Chapter 11

SIMULATION EXAMPLES

11.1 Introduction

It has been emphasized throughout the text that simulation is a problem solving technique, which is applicable in almost every field of decision making. There is a wide variety of its applications, which were briefly discussed in Chapter 1. In each chapter, some examples, drawn from various fields, have been discussed. In this chapter, a few more examples illustrating the application of the technique have been discussed. These include applications of simulation in maintenance and replacement, capacity planning, profit analysis, manufacturing systems, and service systems like car washing system. To make the process of simulation clear, the logic of the model is first explained by doing manual simulation, and then the computer model is given. Some results obtained from computer models have also been included in each case, which help in demonstrating the difference in accuracy of results with the increase in simulation run.

11.2 Simulation of a Counter Service

A coffee house in a busy market operates counter service. The proprietor of the coffee house is faced with the problem of determining the number of bearers he should employ at the counter so that the average waiting time of the customers does not exceed 2 minutes. At present the counter runs with two bearers. There is enough space for the customer to comfortably wait and hardly any customer refuses to enter the system because of waiting line. After recording the data for a number of days, the following frequency distributions of inter-arrival time of customers and the service time at the counter have been established.

Inter-arrival Time (minutes)	*Frequency %*	*Service time (minutes)*	*Frequency %*
0.0	5	1.0	5
0.5	35	2.0	25
1.0	25	3.0	35
1.5	15	4.0	20
2.0	10	5.0	15
2.5	7		
3.0	3		

Demonstrate the technique of simulation by simulating the system for about 30 arrivals for various alternative number of bearers and determine the suitable answer to the problem.

Solution: It is a simple queuing situation, where customers arrive at the counter for taking coffee. Depending upon the number of bearers, the waiting time of the customers will vary. This system can be treated as a single queue multiple channels in parallel system, where the customer at the head of the queue will enter the service channel, which becomes available at the earliest

Let us use two-digit random numbers for generating inter-arrival and service times. The random numbers will be allocated as under :

Inter-arrival time	*Frequency %*	*Cumulative frequency*	*Random numbers*
0.0	5	5	00 to 04
0.5	35	40	05 to 39
1.0	25	65	40 to 64
1.5	15	80	65 to 79
2.0	10	90	80 to 89
2.5	7	97	90 to 96
3.0	3	100	97 to 99

Service time	*Frequency %*	*Cumulative frequency*	*Random numbers*
1.0	5	5	00 to 04
2.0	25	30	05 to 29
3.0	35	65	30 to 64
4.0	20	85	65 to 84
5.0	15	100	85 to 99

From the frequency distribution of inter-arrival times, the mean inter-arrival time can be determined as

$$\overline{\mu} = (5 \times 0.0 + 35 \times 0.5 + 25 \times 1.0 + 15 \times 1.5 + 10 \times 2.0 + 7 \times 2.5 + 3 \times 3.0) / 100$$

$$= 1.115 \text{ minutes}$$

Similarly the mean service time,

$$\overline{\lambda} = (5 \times 1.0 + 25 \times 2.0 + 35 \times 3.0 + 20 \times 4.0 + 15 \times 5.0) / 100$$

$$= 3.15 \text{ minutes.}$$

It can be estimated that the average number of bearers should be around 3. The simulation can be started with 2 bearers and then the number of bearers can be increased by one at a time.

Simulation of the counter system with 2 and 3 bearers for 30 arrivals is given in Table 11.1. The random numbers used have been taken from the random number table. As soon as a customer arrives, he is attended by a bearer who is free at that time. If no bearer is free, customer waits till a bearer becomes available. In case more than one bearer are free, any one can start service on the waiting customer. Since we are interested only in the waiting time of the customers, the idle time of the bearer has not been recorded in the simulation.

In this example, the use of common random numbers has also been demonstrated. Here the performance of the system is being compared with change in number of bearers. Two configurations have been tested, one with two bearers and the second with three bearers. To ensure that the two configurations are tested on the same customers, the inter-arrival times and the service times of the customers have been generated only once, the same times have been used in the two configurations.

The simulation results show that when there are two bearers the average waiting time of customer is 4.6 minutes. When the number of bearers is increased to three the waiting time comes down to 0.267 minutes, hence there is no need of further increasing the number of bearers. Thus, three bearers should be employed.

The computer programme for this simulation is given below:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#define SEED 12345
main()
    {       /* Simulation of a Coffee House with S number of bearers.
            Unlimited waiting area.
             at[] are the arrival times and p[] are the corresponding
             cumulative probabilities.
             t[] are the service times and f[] are the corresponding
             cumulative probabilities.
             counter is the counter of arrivals
             run is the number of arrivals for which simulation is to be
run.
             n is the number of values of at[].
             m is the number of values of t[].
             iat is interarrival time.
             nat is the next arrival time according to simulation clock.
             st is the service time of a customer. */

             int i,j,k,kk,run,counter;
             int n=6,m=5,s=4;
             float se[10];
             float r,iat,st,nat,wt,cwt;

             float p[]={0.,.05,.4,.65,.80,.90,.97,1.0};
             float at[]={0.,.5,1.0,1.5,2.0,2.5,3.0};
             float f[]={0.,.05,.3,.65,.85,1.0};
             float t[] ={1.,2.,3.,4.,5.};

             printf("\n Enter the value of run  ");
             scanf("%d",&run);
          /* Simulation is to be run for servers varying from 2 to 5.*/
             for(j=2;j<=s;++j) {
             srand(SEED);
          /* Initialise the variables*/
             counter=0;nat=0.0;cwt=0.0;
             for(i=1;i<=j;++i){
             se[i]=0.0; }
          /* Run the simulation for prescribed length of run*/
             cwt=0; counter=0;;
             for(counter=1;counter<=run;++counter){
          /* Generate the interarrival and service times for the new
arrival*/
             r=rand()/32768.;
             for(i=0;i<=n;++i) {
             if(r>p[i] && r<=p[i+1]) iat=at[i]; }
```

Table 11.1

						2 Bearer System					3 Bearer System						
Arrival number	Random number	Inter-arrival time	Random number	Service time	Cumulative arrival time	Bearer 1 SB	Bearer 1 SE	Bearer 2 SB	Bearer 2 SE	Customer waiting time	Bearer 1 SB	Bearer 1 SE	Bearer 2 SB	Bearer 2 SE	Bearer 3 SB	Bearer 3 SE	Customer waiting time
1	–	–	31	3	0	0.0	3.0				0.0	3.0					0
2	48	1.0	46	3	1.0			1.0	4.0	0			1.0	4.0			0
3	51	1.0	24	2	2.0	3.0	5.0			1.0					2.0	4.0	0
4	06	0.5	54	3	2.5			4.0	7.0	1.5	3.0	6.0					0.5
5	22	0.5	63	3	3.0	5.0	8.0			2.0			4.0	7.0			1.0
6	80	2.0	82	4	5.0			7.0	11.0	2.0					5.0	9.0	0
7	56	1.0	32	3	6.0	8.0	11.0	–	–	2.0	6.0	9.0					0
8	06	0.5	14	2	6.5			11.0	13.0	4.5			7.0	9.0			0.5
9	92	2.5	63	3	9.0	11.0	14.0			2.0					9.0	12.0	0
10	51	1.0	18	2	10.0			13.0	15.0	3.0	10.0	12.0					0
11	13	0.5	52	3	10.5	14.0	17.0			3.5			10.5	13.5			0
12	65	1.5	82	4	12.0			15.0	19.0	3.0					12.0	16.0	0
13	60	1.0	03	1	13.0	17.0	18.0			4.0	13.0	14.0					0
14	51	1.0	62	3	14.0	18.0	21.0			4.0			14.0	17.0			0
15	50	1.0	22	2	15.0			19.0	21.0	4.0	15.0	17.0					0
16	13	0.5	61	3	15.5	21.0	24.0			5.5					16.0	19.0	0.5
17	94	2.5	29	2	18.0			21.0	23.0	3.0	18.0	20.0					0
18	57	1.0	50	3	19.0			23.0	26.0	4.0			19.0	22.0			0
19	26	0.5	24	2	19.5	24.0	26.0			4.5					19.5	21.5	0
20	78	1.5	69	4	21.0			26.0	30.0	5.0	21.0	25.0					0
21	33	0.5	54	3	21.5	26.0	29.0			4.5					21.5	24.5	0
22	60	1.0	66	4	22.5	29.0	33.0			6.5			22.5	26.5			0
23	31	0.5	46	3	23.0			30.0	33.0	7.0					24.5	27.5	1.5
24	64	1.0	37	3	24.0	33.0	36.0			9.0	25.0	28.0					1.0

25	89	2.0	18	2	26.0			33.0	35.0	7.0			26.5	28.0			0.5
26	64	1.0	68	4	27.0			35.0	39.0	8.0					27.5	31.5	0.5
27	44	1.0	86	5	28.0	36.0	41.0			8.0	28.0	33.0					0
28	33	2.0	37	3	30.0			39.0	42.0	9.0			30.0	33.0			0
29	28	0.5	82	4	30.5	41.0	45.0			10.5					31.5	35.5	1.0
30	71	1.5	48	3	32.0			42.0	45.0	10.0	33.0	36.0					1.0

Total Waiting time = 138.0

$$\text{Average waiting time} = \frac{8.0}{30}$$

$$= 4.6 \text{ min}$$

Total Waiting time = 8.0

$$\text{Average waiting time} = \frac{8.0}{30}$$

$$= 0.27 \text{ min}$$

```
        r=rand()/32768.;
        for(i=0;i<=m;++i) {
        if(r>f[i] && r<=f[i+1]) st=t[i]; }
        nat=nat+iat;
    /* Determine the earliest service ending and the bearer
        and update the statistics. */

        float min=99.9;
        for(i=1;i<=j;++i) {
        if(se[i]<=min) {min=se[i];k=i;} }
        if(nat<=min) {se[k]=min+st;wt=min-nat;}
        else se[k]=nat+st;
        cwt=cwt+wt;  }

        printf("\nServers=%d Total arrivals=%d Average Waiting
time=%6.2f",
         j,counter,cwt/run);
    }
    }
```

An output obtained from this model is given below. The average waiting time of customers, when there are two, three or four bearers, for different values of length of run are given. As the average waiting time goes on decreasing, a longer run is required to get reasonably accurate results.

Run Length	*Average Waiting time per customer*		
30	S = 2	S = 3	S = 4
50	4.25	1.87	1.55
100	9.81	1.07	0.73
200	5.60	0.54	0.61
500	3.06	0.21	0.55

When this simulation is run for 1000 arrivals, with four different values of SEED, the results obtained are,

	S = 2	*S = 3*	*S = 4*	*S = 5*
	2.24	0.11	0.52	0.02
	2.06	0.96	0.87	0.08
	1.98	0.91	0.98	0.01
	2.15	0.90	0.04	0.01
Average Value	2.11	0.72	0.60	0.03

In this simulation experiment, there is a wide variation in the results obtained, for different values of SEED. Thus, a well designed multi-replication program is required to achieve reliable results. Further refinement of this programme is left to the students, as an exercise.

11.3 Simulation of Maintenance and Replacement Problem

A piece of equipment contains four identical vacuum tubes and can function only if all the four are in working order. For each tube the operating time until failure has approximately uniform distribution from 1000 to 2000 hours. The present practice is to replace a tube only when it fails. This results in very frequent shutdowns. An alternative proposal has been made that all the four tubes be replaced whenever any one of them fails.

The equipment must be shut down for one hour to replace one tube or for two hours to replace all the four tubes. The shutdown and replacement of the tubes costs Rs. 200 per hour, while each tube costs Rs. 100.

Simulate the alternative maintenance policies for about 10,000 hours and determine the better of the two.

Solution: The lives of tubes can be generated by using random numbers between 0 and 1. If r is a random number, then life of the tube (y) will be given by

$$y = 1000 + r \times 1000, \quad 0 < r \leq 1$$

The random number can be a two-digit, three-digit or four-digit etc. Larger the number of digits, more accurate will be the simulation, but longer will be the simulation time. Since the simulation is to be conducted by hand calculations, it will be better to use two-digit random numbers, so that the life of the tubes is in 100's of hours. For example, if 0.23, .05, .75 and .43 are the random numbers, the corresponding tube lives are 1230, 1050, 1750 and 1430 hours.

Present Policy

A tube is replaced as and when it fails. The simulation can be started by assuming that all the tubes are new at zero time. The lives of four tubes are generated, whichas shown in Table 11.2, are 1230, 1050, 1750 and 1430 hours respectively. Tube to fail at the earliest is identified, which in this case is T_2 with life of 1050 hours. Clock is advanced by 1050 hours, and time to failure of the other three tubes is updated. Tube T_2 is replaced with a new tube. The next random number is 0.34, giving life of T_2 as 1340 hours. The next tube to fail is T_1 with time to failure of 180 hours. Clock is advanced to 1050 + 180 = 1230 hours, a new life for T_1 is generated and time to failure of other tubes is updated. The process continues, as in Table 11.2.

During the first 10150 hours of operation there had been 26 shutdowns of the system. Since one tube is replaced each time, each shutdown was for one hour.

Shutdown cost = $26 \times 200 = 5200$

Cost of tubes = $26 \times 100 = 2600$

Total cost = Rs. 7800/-.

Proposed Policy

Replace all the four tubes when any one of them fails. Since a comparison of the two policies is to be made, it will be better to use the same sequence of random numbers. If the simulation is run on a computer and for sufficiently long time, then even different sequences of random numbers can be used. Simulation for the proposed policy is run as in Table 11.3. At time zero all the tubes are new. When tube T_2 fails after 1050 hours, all the four are replaced, that is, lives for all the four are generated by using the next four random numbers, and count of shutdown is increased by 1. The tube to fail at the earliest is identified, which at this time is T_1 with life 1340 hours. Clock is advanced by this time, new lives for the tubes are generated, count is updated. The simulation process continues.

During the first 10100 hours there had been 9 shutdowns of 2 hours each and 36 tubes have been replaced.

Shutdown cost = $9 \times 2 \times 200 = 3600$

Cost of tubes = $36 \times 100 = 3600$

Total cost = Rs. 7200/-.

Thus with the small simulation of about 10,000 hours, the proposed policy seems to be better. To have more confidence in the results, it should be run for a longer time, better on a computer.

A computer simulation coded in C language is given below.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
# define SEED  12345
main()
    {
        /* Maintenance and Replacement Problem to evaluate two alternate maintenance
policies. */
        int week=1,t[10],count=0,kont,clock,small,i,k,jj;
        float x;  int run=20000;
        int cost1,cost2; srand(SEED);
        /* Present policy*/
        for (i=1;i<=4;++i) {
        x=rand()/32768.;
        t[i]=(int)(1000.+x*1000.);
        printf(" %8d",t[i]);}
        clock=0;
        printf("\n CLOCK   T1   T2   T3   T4  COUNT");
        while(clock<=run) {
        printf("\n %5d %5d %5d %5d %5d %5d ",clock,t[1],t[2],t[3],t[4],count);

        small=999999;
        for(i=1;i<=4;++i) {
        if(t[i]<small) {small=t[i];jj=i;}

        /*printf("\njj=%d small=%d",jj,small); */
        }
        for(i=1;i<=4;++i) {
        t[i]=t[i]-small; }
        t[jj]=(int)(1000.+(rand()/32768.)*1000.);
        clock=clock+small; count=count+1;
        }
        /* proposed Policy*/
        clock=0,kont=0; srand(SEED);
        printf("\n CLOCK   T1   T2   T3   T4  KONT");
        while(clock<=run) {
        for (i=1;i<=4;++i) {
        x=rand()/32768.;
        t[i]=(int)(1000.+x*1000.);}
        /*printf(" %8d",t[i]);} */
        printf("\n %5d %5d %5d %5d %5d %5d ",clock,t[1],t[2],t[3],t[4],kont);
        small=999999;
```

```
for(i=1;i<=4;++i) {
if(t[i]<small) small=t[i];}
clock=clock+small;  kont=kont+1;
/*printf("\n small=%d clock=%d count=%d",small,clock,count);*/
}
cost1=count*(200+100); cost2=kont*2*200+kont*4*100;
printf("\n Cost Present policy=%d  Cost proposed policy=%d",cost1,cost2);
printf("any digit");
scanf("%d",k);
}
```

Table 11.2

Clock	*Random Numbers*	Time to failure T_1	T_2	T_3	T_4	*Count shutdown*
0	.23,.05,.75,.43	1230	1050	1750	1430	0
0 + 1050 = 1050	.34	180	1340	700	380	1
1050 + 180 = 1230	.90	1900	1160	520	200	2
1430	.72	1700	960	320	1720	3
1750	.46	1380	640	1460	1400	4
2390	.84	740	1840	820	760	5
3130	.66	1660	1100	80	20	6
3150	.09	1640	1080	60	1090	7
3210	.14	1580	1020	1140	1030	8
4230	.59	560	1590	120	10	9
4240	.41	550	1580	110	1410	10
4350	.05	440	1470	1050	1300	11
4790	.40	1400	1030	610	860	12
5400	.96	790	420	1960	250	13
5650	.12	540	170	1710	1120	14
5820	.83	370	1830	1540	950	15
6190	.91	1910	1460	1170	580	16
6770	.46	1330	880	590	1460	17
7360	.53	740	290	1530	870	18
7650	.39	450	1390	1240	580	19
8100	.11	1110	940	790	130	20
8230	.19	980	810	660	1190	21
8890	.43	320	150	1430	530	22
9040	.11	170	1110	1280	380	23
9210	.44	1440	940	1110	210	24
9420	.09	1230	730	900	1090	25
10150	.82	500	1820	170	360	26

Table 11.3

Clock	Random Numbers	Time to failure T_1	T_2	T_3	T_4	Count shutdown
0	.23 .05 .75 .43	1230	1050	1750	1430	0
1050	.34 .90 .72 .46	1340	1900	1720	1460	1
2390	.84 .66 .09 .14	1840	1660	1090	1140	2
3480	.59 .41 .05 .40	1590	1410	1050	1400	3
4530	.96 .12 .83 .91	1960	1120	1830	1910	4
5650	.46 .53 .39 .11	1460	1530	1390	1110	5
6760	.19 .43 .11 .44	1190	1430	1110	1440	6
7870	.09 .82 .90 .25	1090	1820	1900	1250	7
8960	.54 .77 .82 .14	1540	1770	1820	1140	8
10100	.43 .45 .17 .86	1430	1450	1170	1860	9

11.4 Simulation of a Capacity Planning Problem

In a machine shop engaged in doing job work, the number of orders received per day range from 1 to 10. The analysis of previous records shows that the frequency of one order, two orders, three orders etc. is the same as 10%. Each order requires a certain number of machine hours. Analysis of past records is summarized below:

Machine hours (x)	Frequency %
$0 < x \leq 5$	5
$5 < x \leq 10$	6
$10 < x \leq 15$	10
$15 < x \leq 20$	14
$20 < x \leq 25$	16
$25 < x \leq 30$	16
$30 < x \leq 35$	13
$35 < x \leq 40$	10
$40 < x \leq 45$	7
$45 < x \leq 50$	3

If an order has to wait it incurs a penalty of Rs. 25 per machine hour per day and if the machines are idle, they incur a cost of Rs. 100 per hour of capacity idle time. The problem is to determine the capacity of the machine shop, so that the cost incurred due to idle capacity and orders waiting is minimum. It can be assumed that the orders received on one day are processed, say, the next day, and no overtime is allowed.

Solution: The number of orders received per day range from one to ten with equal probability. If one-digit random numbers are used to simulate the orders, the random number will directly give the number of orders as 1, 2, 3, ..., 9, and when the random number is 0, it will be taken as 10 orders.

The machining hours required for the orders vary from zero to fifty, according to the given frequency distribution. The cumulative frequency curve for machine hours per order is given in Fig. 11.1. Since the frequency distribution is continuous, the machining time for an order will be generated by using this curve. For example, corresponding to random number 17, the machining hours are 13.5 and corresponding to random number 49, the machining hours are 24.5.

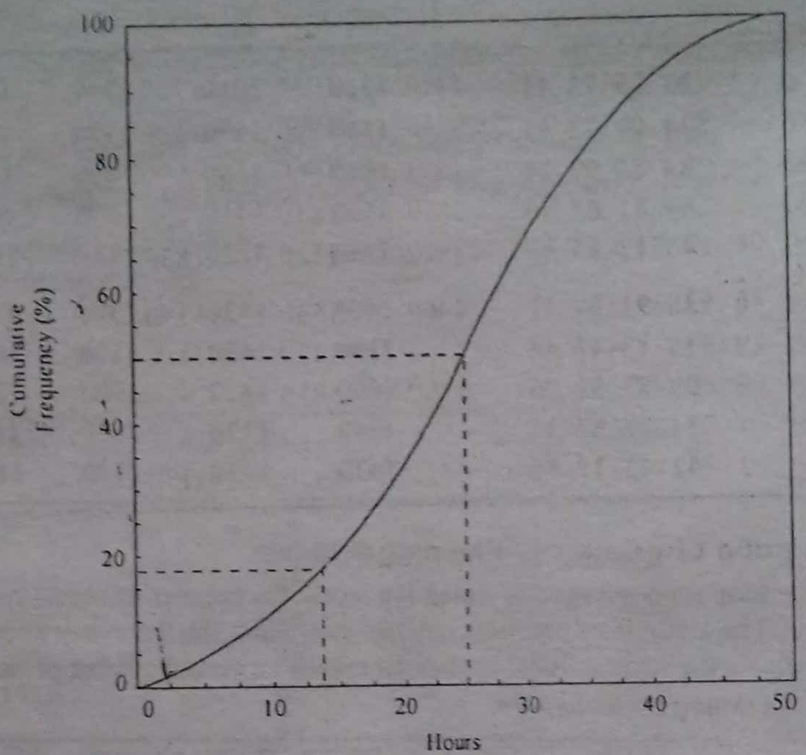

Fig. 11.1

Table 11.4

Machine Hours	*Cumulative frequency %*
$0 < x \leq 5$	5
$5 < x \leq 10$	11
$10 < x \leq 15$	21
$15 < x \leq 20$	35
$20 < x \leq 25$	51
$25 < x \leq 30$	67
$30 < x \leq 35$	80
$35 < x \leq 40$	90
$40 < x \leq 45$	97
$45 < x \leq 50$	100

The objective in the present situation is to determine the machining capacity of the shop, so as to minimise the orders waiting and idle capacity costs.

Search Procedure: The optimum value of capacity can be determined by a search procedure. The search is started with an arbitrary value of capacity or the value determined by some approximate method. Let it be H hours. The simulation is run for sufficiently long time, the waiting time of orders and idle machining capacity are determined, from which the cost is computed. The capacity is then increased by some amount, say, DH. Simulation is again run, if the cost decreases, capacity is further increased and if the cost increases, capacity is decreased by a step DH′, which may be less than half of DH. The process continues, till the near optimum capacity is obtained.

Alternatively, costs corresponding to some values of capacity are determined, and a curve is plotted, from which the minimum cost capacity can be determined.

In the present case, average load or the average demand can be determined by generating the demand for some days. In Table 11.5, number of orders and their machining hours have been generated for 20 days.

Total hours for 20 days = 2823.5

Average Machining load = 141.175 hours/day

We can start the search with machine shop capacity of 140 hours. The simulation of the system for 20 days is shown in Table 11.6. On first day, there are two orders requiring 38 machine hours and hence the shop is idle for 140 – 38 = 102 hours. On 2nd day the orders are worth 147.5 hours and hence orders worth 147.5 – 140 = 7.5 hours wait for the day, and are carried to the third day. On third day there are orders worth 138 hours and orders worth 7.5 hours are already in the queue, making the total machining load for the day 138 + 7.5 = 145.5 hours.

At the end of 20 days,

Total idle capacity = 178.0 hours

and total waiting time of orders = 1820.5 hours

cost of idle capacity = 178.0 × 100 = 17800

cost of waiting time = 1820.5 × 25 = 45500

Total cost = Rs. 63,300.

Now the capacity of the shop is incremented to, say, 150. Again the simulation is run. Here, the number of orders and their machining hours have been kept the same in each simulation, to avoid the error due to randomness. Simulation is given in Table 11.7.

Total idle capacity = 258.0 hours

Total waiting time = 962.5 hours

Total cost = 25800 + 24062.5

= Rs. 49,862.5.

Since the cost has decreased, increment the capacity to 160 hours a day. Simulation is given in Table 11.8.

Total idle capacity = 427.0 hours

Total waiting time = 678.5 hours

Total cost = 42700 + 16962.5

= Rs. 59,662.5.

Table 11.5

Days	*No. of orders*	*Random Numbers*	*Machine hours*	
1	2	17, 49	13.5 + 24.5 =	38.0
2	7	70, 15, 29, 04, 91 23,50	31.0 + 12.5 + 18 + 4.5 + 41 + 15.75 + 24.75 =	147.5
3	5	70, 64, 49, 00, 03	31 + 29 + 24.5 + 50 + 3.5 =	138.0
4	2	56, 72	26.5 + 32 = 58.5	
5	8	06, 74, 26, 87, 97, 99 79, 85	6 + 32.5 + 17 + 38.5 + 45 + 49 + 34.5 + 37.5 =	260.0
6	4	28, 86, 10, 55	18 + 38 + 9.5 + 26 =	91.5
7	4	63, 00, 94, 98	29 + 50 + 43 + 47 =	169.0
8	1	91	40.5	40.5

9	10	250.0
10	3	71.0
11	5	172.5
12	6	150.5
13	8	105.0
14	10	230.5
15	4	210.0
16	2	100.0
17	7	60.5
18	9	170.5
19	6	235.0
20	4	125.0
	Total =	2823.5

The total cost has increased from the previous cost, and hence decrease the capacity to say 155 hours. The total cost corresponding to this capacity value is Rs. 53,750, which is less than the previous value and hence continue the search by decreasing the capacity further to say 152.5 hours, 147.5 hours etc., till the optimum capacity is obtained. Some values of costs obtained are given below.

Capacity	*Total cost (Rs.)*
140.0	63,300.0
145.0	55,712.5
147.5	53,049.0
150.0	49,862.5
152.5	52,462.0
160.0	59,662.5

The capacity cost curve for the given system is shown in Fig. 11.2. The minimum cost corresponds to machining capacity of around 150 hours per day.

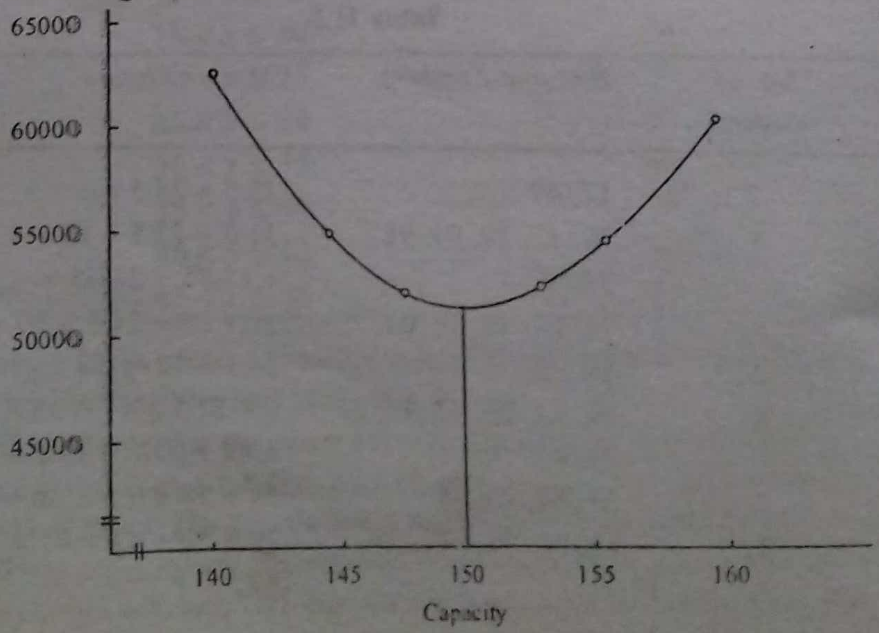

Fig. 11.2

Table 11.6: Machining Capacity 140 hours per day

Day	*Demand*	*Capacity idle*	*Orders Waiting*
1	38.0	140 – 38.0 = 102.0	
2	147.5	—	147.5 – 140 = 7.5
3	138.0 + 7.5	—	145.5 – 140 = 5.5
4	58.5 + 5.5	140 – 64.0 = 76.0	
5	260.0	—	260 – 140 = 120.0
6	91.5 + 120.0	—	211.5 – 140 = 71.5
7	169.0 + 71.5	—	240.5 – 140 = 100.5
8	40.5 + 100.5	—	141.0 – 140 = 1.0
9	250.0 + 1.0	—	251.0 – 140 = 111.0
10	71.0 + 111.0	—	182.0 – 140 = 42.0
11	172.5 + 42.0	—	214.5 – 140 = 74.5
12	150.5 + 74.5	—	225.0 – 140 = 85.0
13	105.0 + 85.0	—	190.0 – 140 = 50.0
14	230.5 + 50.0	—	280.5 – 140 = 140.5
15	210.0 + 140.5	—	350.5 – 140 = 210.5
16	100.0 + 210.5	—	310.5 – 140 = 170.5
17	60.5 + 170.5	—	231.0 – 140 = 91.0
18	170.5 + 91.0	—	261.5 – 140 = 121.5
19	235.0 + 121.5	—	356.5 – 140 = 216.5
20	125.0 + 216.5	—	341.5 – 140 = 201.5
		Total = 178.0	Total = 1820.5

Table 11.7: Machining Capacity 150 hours a day

Day	*Demand Machine hours*	*Capacity idle*	*Orders Waiting*
1	38.0	150 – 38.0 = 112.0	—
2	147.5	150 – 147.5 = 2.5	—
3	138.0	150 – 138.0 = 12.0	—
4	58.5	150 – 58.5 = 91.5	—
5	260.0	—	260 – 150 = 110.0
6	91.5 + 110.0	—	201.5 – 150 = 51.5
7	169.0 + 51.5	—	220.5 – 150 = 70.5
8	40.5 + 70.5	150 – 111 = 39.0	—
9	250.0	—	250.0 – 150 = 100.0
10	71.0 + 100.0	—	171.0 – 150 = 21.0
11	172.5 + 21.0	—	193.5 – 150 = 43.5
12	150.5 + 43.5	—	194.0 – 150 = 44.0
13	105.0 + 44.0	150 – 149.0 = 1.0	—
14	230.5	—	230.5 – 150 = 80.5

15	210.0 + 80.5	—	290.5 – 150 = 140.5
16	100.0 + 140.5	—	240.5 – 150 = 90.5
17	60.5 + 90.5	—	151.0 – 150 = 1.0
18	170.5 + 1.0	—	171.5 – 150 = 21.5
19	235.0 + 21.5	—	256.5 – 150 = 106.5
20	125.0 + 106.5	—	231.5 – 150 = 81.5
		Total = 258.0	Total = 962.5

Table 11.8: Machining Capacity 160 hours per day

Day	*Demand Machine hours*	*Capacity idle*	*Orders Waiting*
1	38.0	160 – 38.0 = 122.0	—
2	147.5	160 – 147.5 = 12.5	—
3	138.0	160 – 138.0 = 22.0	—
4	58.5	160 – 58.5 = 101.5	—
5	260.0	—	260 – 160 = 100.0
6	91.5 + 100.0	—	191.5 – 160 = 31.5
7	169.0 + 31.5	—	200.5 – 160 = 40.5
8	40.5 + 40.5	160 – 81.0 = 79.0	—
9	250.0	—	250.0 – 160 = 90.0
10	71.0 + 90.0	—	161.0 – 160 = 1.0
11	172.5 + 1.0	—	173.5 – 160 = 13.5
12	150.5 + 13.5	—	164.0 – 160 = 4.0
13	105.0 + 4.0	160 – 109.0 = 51.0	—
14	230.5	—	230.5 – 160 = 70.5
15	210.0 + 70.5	—	280.5 – 160 = 120.5
16	100.0 + 120.5	—	220.5 – 160 = 60.5
17	60.5 + 60.5	160 – 121.0 = 39.0	
18	170.5	—	170.5 – 160 = 10.5
19	235.0 + 10.5	—	245.5 – 160 = 85.5
20	125.0 + 85.5	—	210.5 – 160 = 50.5
		Total = 427	Total = 678.5

The results obtained from this simulation of 20 days can by no means be considered as reliable, firstly because of a very small simulation run and secondly because of the initial bias of starting the system empty. The exercise carried out, however, is a good demonstration of the technique of simulation employing common random numbers.

The computer program for this simulation is given below:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
```

```
{  /* CAPACITY PLANNING PROBLEM ,
   f[]        Cumulative frequency
   ords       Number of orders
   days       A variable for days
   run        Days for which simulation is to be run(5000)
   capacity   Available capacity in machine hours
   chours     Cumulative hours
   idle       Idle capacity on a day
   cidle      Cumulative idle capacity
   wtng       Orders waiting
   cwtng      Cumulative Orders waiting
   costc      Cost per hour of idle capacity time
   costw      Cost per hour of waiting orders
   cost       Total cost  */

   float f[]= {.0,.05,.11,.21,.35,.51,.67,.80,.90,.97,1.00};
   int i,j,k,m,n,ords,days,run;
   float
r,hrs,hours,chours=0.,capacity,idle,wtng,cidle=0.,cwtng=0.;
      float cost,costc=50.,costw=100.;
      capacity=190.;

  /* for(n=0;n<=10;++n) {printf("\n %6.2f",f[n]);}  */
     printf("\n Enter run");
     scanf("%d",&run);
     printf("\n Idle capacity cost=%5.2fper hour,",costc);
     printf("\n Waiting time cost =%5.2fper hour",costw);
     while(capacity<=230.){
     chours=0.;cidle=0.;cwtng=0.;
     hours=0.;
  /* printf("\n    days    orders   Hours    idle    wtng");*/
     for(days=1;days<=run;++days) {
     /*generate orders*/
     r=rand()/32768.;
     for(i=0;i<=9;++i) {
     j=i+1;
     if(r>i/10.&& r<=j/10.) ords=j; }
     k=ords;
     /*generate machining hours for orders*/
     for(j=1;j<=k;++j) {
     r=rand()/32768.;
     for(i=0;i<=10;++i) {
     if(r>f[i] && r<=f[i+1]) { hrs=i*5+r*5;
  /* printf("\n%6.4f %7.2f",r,hrs);*/}}
     hours=hours+hrs; }
```

```
        if(capacity>= hours) {idle= capacity-hours; cidle=cidle+idle;}
        else {wtng=hours-capacity;
        cwtng=cwtng+wtng;}
    /*   chours=chours+hours;   */
    /*printf("\n %5d   %5d %9.2f %6.2f
%6.2f",days,ords,hours,idle,wtng);*/
        hours=wtng;
        idle=0.; wtng=0.;
        }
    /*   printf("\n days=%d cidle=%7.2f cwtng=%7.2f
chours=%7.2f",run,cidle,cwtng,chours);
    */
        cost=cidle*costc+cwtng*costw;
        printf("\n Capacity=%6.2f Total cost=%6.2f Cost per
day=%6.2f",
            capacity,cost,cost/days);

        capacity=capacity+5.;

        }}
```

The cost per day obtained for costs of Rs. 100.00 per hour for idle capacity of machines and Rs. 25.00 per hour of waiting time of orders, for a simulation run of 5000 days is given below:

Capacity	*Cost per day*
150	5600.07
155	4737.93
160	4310.01
165	4701.05
170	4879.35

The optimum capacity of the machine shop for the given situation is 160 machine hours a day. When the simulation run is increased the optimum capacity fluctuates between 160 and 165.

When the cost of idle capacity and cost of waiting time is made equal and kept at Rs. 100/- per hour, the optimum capacity comes to around 190. And when the cost of waiting time is Rs. 100/- per hour and cost of idle capacity Rs. 50/- per hour, the value of optimum capacity rises to around 205.

It is left as an exercise to the students to conduct a pilot experiment and to design the simulation run.

11.5 Simulation of a Profit Analysis Problem

A firm is considering the introduction of a new product, and knows with reasonable confidence that the fixed cost will be Rs. 60,000, while the selling price will have to be kept at Rs. 10 due to competition in the market. There is uncertainty about the variable costs and the product demand. It is estimated that the variable cost will be between Rs. 4.5 and Rs. 5.5. The demand for the product will depend upon the reaction of its competitors. If they react strongly in the first year, sales are expected to be 10,000 to 12,000 units and if the reaction is not strong, sales could be 13,000 to

15,000 units in the first year. The firm believes that there are 60% chances of their competitors reacting strongly. The firm likes to ascertain its chances of breaking even in the first year of sales. What are the chances of the loss being more than Rs. 5,000?

Solution: In this case, the management of the firm likes to have some idea of the risk involved in introducing its new product. For a year, the profit of the firm can be computed as,

$$\text{Profit} = \text{Demand} \times (\text{S.P} - \text{V.C}) - \text{F.C}$$

where S.P, V.C and F.C stand for selling price, variable cost and fixed cost respectively.

The fixed cost and selling price are constant at Rs. 60,000 per year and Rs. 10 per unit. The demand and variable costs are to be generated for each year.

Assuming that the variable cost is uniformly distributed over the range 4.5 to 5.5, the cost can be generated as

$$VC = 4.5 + r\,(5.5 - 4.5)$$

where r is a random number between 0 and 1. To keep the computations simple, only two-digit random numbers will be used.

The demand of the product depends upon the competitors' reactions, which can be strong or weak, with 60% chances of being strong. If we use one-digit random numbers to generate the reaction, then the reaction is strong if the random digit is 0, 1, 2, 3, 4 or 5 and weak, if it is 6, 7, 8 or 9.

After knowing the reaction of competitors, demand can be generated by assuming the demand to be say uniformly distributed over the given ranges.

For strong reaction : 10,000 to 12,000 units

For weak reaction : 13,000 to 15,000 units.

Drawing random numbers between 0 and 1, demand,

For strong reaction $= 10{,}000 + r\,(12000 - 10000)$

For weak reaction $= 13{,}000 + r\,(15000 - 13000)$

To keep the calculations simple only two-digit random numbers will be used. Once the variable cost and demand for the year is determined, the profit can be computed. For determining the probability of breaking even, the experiment will have to be repeated a number of times. Larger the number of trials, more accurate will be the result. Here we will simulate the system for 30 trials, based on which the length of run can be estimated, and then the experiment should be run on a computer for the estimated length of run. Simulation of first 30 trials is given in Table 11.9.

From the 30 observations in Table 11.9, it can be observed that 13 times, the profit is negative and 17 times, it is positive. Thus, the probability of breaking even in the first year of production is 56.67%.

The frequency distribution of the profits is as follows :

Profit Range			*Frequency*
20000	to	25000	1
15000	to	20000	3
10000	to	15000	5
5000	to	10000	3
0	to	5000	5
0	to	− 5000	7
− 5000	to	− 10,000	5
− 10000	to	− 15,000	1

Out of 30 observations, six times the loss is more than 5000 or there are 20% chances of loss being greater than 5000. Similarly, there are 12/30 × 100 = 40% chances of profit being more than 5000.

Length of Simulation Run:

The length of simulation run can be determined by computing the standard deviation of the profits. Either the 30 observations of Table 11.9 can be analysed for determining the standard deviation, or grouped observations in the above frequency distribution can be analysed. The analysis of 30 observations will give more accurate result.

Using the frequency distribution

$$\bar{x} = \frac{\Sigma f \cdot x}{\Sigma f} = \frac{1{,}05{,}000}{30} = 3{,}500$$

$$\sigma = \sqrt{\frac{\Sigma(\bar{x}-x)^2 f}{n}} = \sqrt{\frac{1575 \times 10^6}{30}} = 7245.7 \approx 7250 \text{ say.}$$

Table 11.9

Trial number	*Random number*	*Variable cost*	*Random number*	*Reaction*	*Random number*	*Demand*	*Profit*
1	.73	5.23	3	S	.72	11440	– 5431.2
2	.55	5.05	9	W	.69	14380	11181.0
3	.32	4.82	3	S	.48	10960	– 3227.2
4	.50	5.00	4	S	.55	11100	– 4500.0
5	.19	4.69	1	S	.17	10340	– 5094.6
6	.44	4.94	2	S	.94	11880	112.8
7	.13	4.63	6	W	.45	13900	14643.0
8	.48	4.98	7	W	.82	14640	13492.8
9	.16	4.66	3	S	.81	11620	2050.8
10	.38	4.88	2	S	.73	11460	– 1324.8
11	.64	5.14	4	S	.53	11060	– 6248.4
12	.62	5.12	7	W	.79	14580	11150.4
13	.68	5.18	9	W	.32	13640	5744.8
14	.20	4.70	4	S	.67	11340	102.0
15	.00	4.50	6	W	.39	13780	15790.0
16	.87	5.37	1	S	.60	11200	– 8144.0
17	.09	4.59	9	W	.92	14840	20284.0
18	.66	5.16	8	W	.23	13460	5146.4
19	.73	5.23	6	W	.16	13320	3536.4
20	.69	5.19	3	S	.99	11980	– 2376.2
21	.63	5.13	0	S	.97	11940	– 1852.2
22	.89	5.39	1	S	.70	11400	– 7446.0
23	.14	4.64	7	W	.69	14380	17076.8
24	.40	4.90	6	W	.42	13840	10584.0
25	.75	5.25	8	W	.31	13620	4695.0
26	.24	4.74	0	S	.78	11560	805.6
27	.13	4.63	7	W	.84	14680	18831.6
28	.83	5.33	1	S	.11	10220	– 12272.6
29	.19	4.69	3	S	.24	10480	– 4351.2
30	.16	4.66	5	S	.21	10420	– 4357.2

If we want the expected value of the profit to be within 500 at 95% confidence level, then the sample size n is,

$$n = \left(\frac{1.96 \times 7250}{500}\right)^2 \doteq 807.69 = 810 \text{ say.}$$

Thus the simulation should be run for 810 trials.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
    {
        /* Profit Analysis Problem
        fc     Fixed cost.
        sp     selling price.
        Variable cost is between vc1 and vc2.
        If the reaction of the opponent is strong, sales could
        be sale1 to sale2 and if the reaction is weak the sales
        could be sale3 and sale4.
        csr    Chances of strong reaction.    */
        int i,n,count=0,kont=0;
        float r,profit,demand,vc,fc=60000.,sp=10.,csr=0.6;
        float vc1=4.5,vc2=5.5;
        float sale1=10000.,sale2=12000.,sale3=13000.,sale4=15000.;
        printf("\n Value of run");
        scanf("%d",&n);
        for(i=1;i<=n;++i) {
        r=rand()/32768.;
        vc=vc1+r*(vc2-vc1);
        r=rand()/32768.;
        if(r<=csr)  demand=sale1+r*(sale2-sale1);
        else demand=sale3+r*(sale4-sale3);
        profit=demand*(sp-vc)-fc;

        if(profit<-5000.) count=count+1;
        if(profit>5000.) kont=kont+1;
          /* printf("\n profit=%7.2f count=%d
kont=%d",profit,count,kont);
                        */      }
        printf("\n Number of trials=%d",n);
        printf("\n Chances of LOSS being greater than
5000=%6.2f%",count*100./n);
        printf("\n Chances of PROFIT being greater than
5000=%6.2f%",kont*100./n);
        }
```

When the program is run for the designed simulation run of 810 trials, the results obtained are :

Chances of loss being greater than 5000 = 43.09%

Chances of profit being greater than 5000 = 37.90%

Thus the chances of breaking even are less than 50%.

11.6 Simulation of an Inventory Problem

Bhangu, the owner of an electronic goods shop, likes to experiment with his inventory policy, to determine the order quantity and reorder points for his goods. He has very good record of the data on the inventory of TV sets, which is given below.

Order cost = Rs. 100 / order

Holding cost = Rs. 5 / set / week

Demand distribution:

Units :	0	1	2	3	4
Probability :	.20	.50	.15	.10	.05

Lead Time distribution :

Weeks :	1	2	3	4	5
Probability :	0.10	0.25	0.50	0.10	0.05

He wants to examine the order quantities ranging from 4 to 10 and reorder points from 3 to 6.

Solution: In this inventory problem, a very large number of combinations of order quantities and reorder points will have to be experimented to reach the best policy. There are 7 levels of order quantity (4 to 10) and four levels (3, 4, 5 & 6) of reorder point, giving $7 \times 4 = 28$ combinations. Alternatively, starting with one combination, variables be changed, so that the experiment progresses in the direction of optimum combination only. A search procedure will have to be developed for this. Since this all is not possible to demonstrate with hand calculations, it is left to the readers to do it on a computer. Here, the simulation of the inventory system for one combination will only be demonstrated.

Both the demand and lead time are discretely distributed. We will use a table of random numbers and apply Monte Carlo method for generating the demand and lead times. The random numbers will be allocated as in Tables 11.10 and 11.11.

Table 11.10. Weekly demand

Units	*Probability*	*Cumulative probability*	*Random numbers*
0	.20	.20	0000 to .2000
1	.50	.70	.2001 to .7000
2	.15	.85	.7001 to .8500
3	.10	.95	.8501 to .9500
4	.05	1.00	.9501 to .9999

Table 11.11. Lead Time

Weeks	*Probability*	*Cumulative probability*	*Random numbers*
1	.10	.10	0000 to .1000
2	.25	.35	.1001 to .3500
3	.50	.85	.3501 to .8500
4	.10	.95	.8501 to .9500
5	.05	1.00	.9501 to .9999

Flow diagram of the simulation model is given in Fig. 11.3 and simulation for 30 weeks time is given in Table 11.12. To start the simulation it is assumed that 8 units are there in the shop. The order quantity is 5 and the reorder point is taken as 3. Whenever the inventory in hand falls to 3 or less, 5 units are ordered. The shortages are assumed to be lost. Orders are placed at the end of the week and received in the beginning of the week.

Over the 30 weeks period

Number of orders placed = 6

Thus, ordering cost = Rs. 600

Holding cost = Rs. 345

Total cost = Rs. 945

Orders lost = 14 units.

This simulation of 30 weeks is too small to give any reliable results. But the trend is clear that, the ordering quantity should be increased and also the reorder point be raised to reduce the shortages.

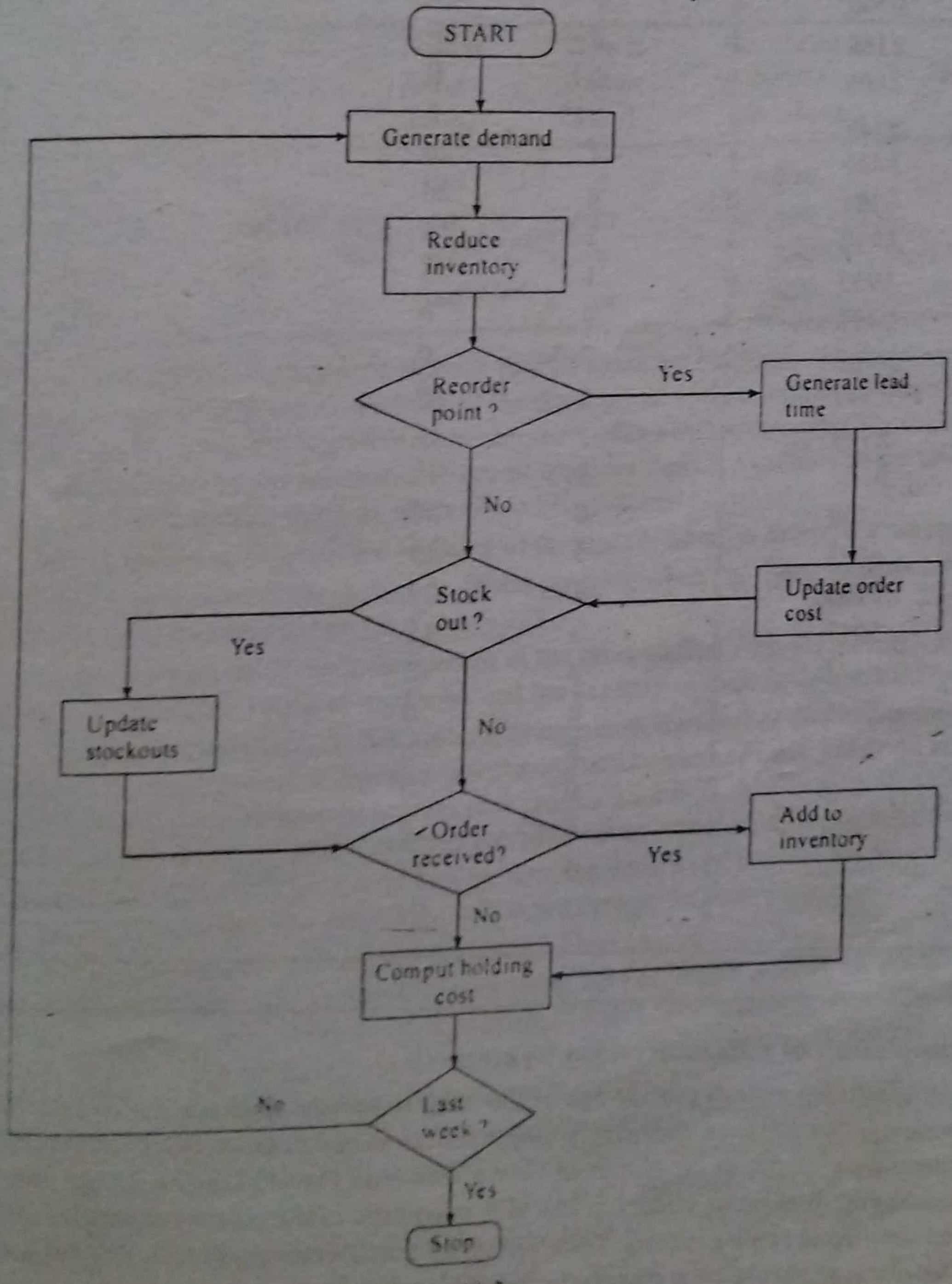

Fig. 11.3

Table 11.12: Inventory Simulation

Week	*Random Number*	*Demand (units)*	*Ending inventory*	*Holding cost*	*Random number*	*Lead time (weeks)*	*Sale Lost*
			8				
1	.8523	3	5	25			
2	.5215	1	4	20			
3	.4450	1	3	15	.8074	3	
4	.0321	0	3	15			
5	.1554	0	3	15			
6	.7612	2	1+5	5			
7	.9706	4	2	10	.6753	3	
8	.8368	2	0	0			
9	.4805	1	0	0			1
10	.9739	4	0+5	0			4
11	.3426	1	4	20			
12	.0281	0	4	20			
13	.8830	3	1	5	.7179	3	
14	.1953	0	1	5			
15	.7497	2	0	0			1
16	.7362	2	0+5	0			2
17	.4237	1	4	20			
18	.2378	1	3	15	.0163	1	
19	.5144	1	2+5	35			
20	.2309	1	6	30			
21	.5863	1	5	25			
22	.6167	1	4	20			
23	.5011	1	3	15	.9745	5	
24	.6423	1	2	10			
25	.9646	4	0	0			2
26	.3454	1	0	0			1
27	.7832	2	0	0			2
28	.2461	1	0+5	0			1
29	.6070	2	3	15	.8639	4	
30	.6039	2	1	05			
				345			14

11.7 Simulation of a Manufacturing System

Manufacturing systems provide one of the most important applications of system simulation. System simulation has been extensively employed as an aid in the analysis and optimization of manufacturing systems, in the design of new production facilities, in the design and layout of warehouses and distribution centres. It has also been used to evaluate the alternative strategies of modifications in the existing systems. Since variety of manufacturing systems is very large, the variety of applications of simulation in manufacturing is also very large.

An example of modelling a manufacturing system, comprising a number of workstations arranged in series, each workstation subjected to breakdowns, is given here. This example is only to illustrate the use of simulation in observing the behaviours of an existing system. Otherwise a large number of simulation packages are available to model various types of situations. The simulation packages are generally specific to the situations and their discussion is beyond the scope of this book.

Example: In a sheet metal shop, sheets are sequentially processed through four presses; the Shear, the Punch, the Form and the Bend Press. The machines being automatic have constant processing times, which are given as processing rates as (sheets/min). Each machine is subject to failures. The times between failures are random and follow exponential distribution. The mean times to failure are given in the table below. The repair times are again random, and have been assumed to be uniformly distributed, with their mean values shown in table and a half width of 5 minutes. The die on each machine is changed after a fixed number of sheets are processed. The time taken to change a die has been assumed to be 25 minutes.

Press	*Process rate (per min.)*	*Time to failure (min)*	*Time to Repair (min)*	*No. sheets to a die change (no. sheets)*	*Time to change die (min)*
Shear	4.5	100	8	500	25
Punch	5.5	90	10	400	25
Form	3.8	180	9	750	25
Bend	3.2	240	20	600	25

In between the workstations, there is limited space for holding work-in-process. An unlimited supply of material is available in front of the shear press, and an unlimited space is available after the Bend press. Each machine processes one sheet after another continuously except when there is no space available for the processed sheet, or when no sheet is available in the upstream buffer, or when machine is broken down or when die is being changed.

It is required to simulate the working of the system, so as to determine the following:

(*i*) Throughput of the system, as sheets per hour, when the capacity of each in-process buffer is 1, 5 and 10 sheets.

(*ii*) The built up of work-in-process in the three buffers with the passage of time, when the buffers are empty to start with and the capacity of each is 10 sheets.

Simulation Procedure : In this manufacturing system, the number of events occurring is quite large and hence the fixed increment time flow mechanism will be more suitable. Since we are interested only in throughput rate and the average amount of in-process stock held, the idle time of machines, the waiting time of sheets, the number of die changes etc. need not be recorded. After each increment in time, the state of all workstations is examined turn by turn, the statistics of interest are recorded and the values of variables are updated. The simulation will move through the following steps :

1. The variables are defined and initialized. Flags like 'down', 'up', 'ddown', 'dup' are defined to indicate the state of the machines being up, *i.e.*, working or down and the state of the die on each machine being worked or being changed. The counters for recording the number of sheets in the in-process buffers and output from the last stage are set to zero. Clock is initialized to zero and the length of run is specified.
2. The starting conditions are decided and accordingly, values are given to the variables. The value of time increment is fixed. Times to failure of machines are computed. Times to die changes are initialized. It is assumed that all machines have a work piece and all start simultaneously as soon as the simulation clock starts. All buffers are taken to be empty.

3. After every increment in time, while the clock is less than the value of run, the status of each workstation is checked as in following steps.
4. The shear press is checked if it is working (not down), time to failure is updated. If it is over, machine is flagged as 'down', time to repair is computed. If it is already down, time to complete repair is updated, and in case it is over, the shear press is flagged up.
5. The state of die of shear press is examined. If it is not down (ddown), the time to die change is updated. If it is over, the machine is put die down (ddown) and the time to change die is computed. If the die is being changed (stage ddown), time to change die is updated, and if over, die is flaged up (dup).
6. For updating the state of processing, it is examined if the shear is up, die is up, and there is space in downstream buffer. If so, the processing time is checked, if it is over, the sheet is added to the downstream buffer, processing time to complete next processing is computed, cumulative buffer is updated.
7. Steps 4, 5 and 6 are repeated for the next workstation that is Punch Press. The only difference is that while updating the state of processing, in addition to space in the downstream buffer, availability of material in the upstream buffer is also checked. Both the upstream and down stream buffers are updated.
8. As in step 7, the procedure is repeated for the Form press.
   At this workstation, the only change is that, there is no restriction on the downstream buffer.
9. When the clock reaches the specified length of run, the results are computed and printed.
   A complete C-program for the simulation of above manufacturing system is given below:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
   {  /*          A MANUFACTURING SYSTEM
      k : The number of parallel washing stalls.
      up: The state when a workstation is not broken down.
      down: When a workstation is broken down and under repair.
      dup: Die of a machine is not being changed.
      ddown: Die of a machine is being changed.
      b1,b2,b3:Number of sheets in in process buffers.
      cb1,cb2,cb3:Cumulative values of b1,b2,b3.
      maxb1,maxb2,maxb3: Capacities of in process buffers.
      Sheet1: Number of sheets to be processed before a die change.
      Stage : A workstation.
      Counter: The counter of sheets output from the last stage.
      kont: The number of observations of state of the systems.
      clock: The record of elapsed time.
      run: The time for which simulation is to be run(length of run).
      rt: Repair time. ttr1: Time to repair machine1.
      ut: time to failure.
      ttf1: Time to failure of machine 1.
      ttcd1: Time to change die 1.
      se1: Time to end processing of the current sheeton machine 1.
       */
      int k,up=999,down=777,dup=888,ddown=666;
```

```
int b1=0,b2=0,b3=0,maxb1=1,maxb2=1,maxb3=1;
long int cb1=0,cb2=0,cb3=0;
int sheet1=500,sheet2=400,sheet3=750,sheet4=600;
int stage1=22,stage2=22,stage3=22,stage4=22;
int counter=0,shear=up,punch=up,form=up,bend=up,shdie=dup,
    punchdie=dup,formdie=dup,bendie=dup;
float r,kont,clock,run,delt,rt,ttr1,ttr2,ttr3,ttr4;
float ut=0.,ttf1,ttf2,ttf3,ttf4,ttcd1,ttcd2,ttcd3,ttcd4;
float se1=4.0,se2=5.0,se3=5.0,se4=6.0;
clock=0.;  delt=1.0;  kont=0.;

printf("\n Length of run=");
scanf("%f",&run);

r=rand()/32768.; ttf1=-100.*log(1-r);
r=rand()/32768.; ttf2=-90.*log(1-r);
r=rand()/32768.; ttf3=-180.*log(1-r);
r=rand()/32768.; ttf4=-240.*log(1-r);
   /*printf("\n %6.2f %6.2f %6.2f %6.2f",ttf1,ttf2,ttf3,ttf4);*/

while(clock<=run) {
/* Check status of shear machine*/
 if(shear==up && ttf1<=clock) {
      shear=down;
      r=rand()/32768.;
      rt=3.+10.*r; ttr1=clock+rt; }
 else if(shear==down && ttr1<=clock) {
      shear=up;
      r=rand()/32768.;
      ut=-100.*log(1-r);/*printf("%6.2f",ut);   */
      ttf1=clock+ut; }
 /* Check the status ofsheer die*/
 if(shdie==dup && sheet1<=0) {
      shdie=ddown;
      ttcd1=clock+25.; }
else if(shdie==ddown && ttcd1<=clock) {
      shdie=dup;
      sheet1=500; }
 /* Check the processing time and buffer*/
 if(b1<maxb1 && shear==up && shdie==dup) {
       if(se1<=clock && b1<=maxb1) {
       b1=b1+1;
       se1=clock+4.5;
       sheet1=sheet1-1;} }

 /*   printf("\n %6.2f %6.2f %6.2f %6.2f
%6.2f",clock,rt,ttr1,ut,ttf1);
     printf("          %5d %5d %6.2f",b1,sheet1,se1);   */
     /* Check status of Punch machine*/
```

```
if(punch==up && ttf2<=clock) {
      punch=down;
      r=rand()/32768.;
      rt=5.+10.*r; ttr2=clock+rt; }
else if(punch==down && ttr2<=clock) {
      punch=up;
      r=rand()/32768.;
      ut=-90.*log(1-r);/*printf("%6.2f",ut);  */
      ttf2=clock+ut; }
/* Check the status ofpunch die*/
if(punchdie==dup && sheet2<=0) {
      punchdie=ddown;
      ttcd2=clock+25.; }
else if(punchdie==ddown && ttcd2<=clock) {
      punchdie=dup;
      sheet2=400; }
/* Check the processing time and buffer*/
if(b2<maxb2 && b1>0 && punch==up && punchdie==dup) {
       if(se2<=clock && b2<=maxb2) {
       b2=b2+1;   b1=b1-1;
       se2=clock+5.5;sheet2=sheet2-1;} }

 /*  printf("\n %6.2f %6.2f %6.2f %6.2f
%6.2f",clock,rt,ttr2,ut,ttf2);
    printf("          %5d %5d %5d %6.2f",b1,b2,sheet2,se2); */

    /* Check status of FORM machine*/
if(form==up && ttf3<=clock) {
      form=down;
      r=rand()/32768.;
      rt=4.+10.*r; ttr3=clock+rt; }
else if(form==down && ttr3<=clock) {
      form=up;
      r=rand()/32768.;
      ut=-180.*log(1-r);/*printf("%6.2f",ut);  */
      ttf3=clock+ut; }
/* Check the status ofForm die*/
if(formdie==dup && sheet3<=0) {
      formdie=ddown;
      ttcd3=clock+25.; }
else if(formdie==ddown && ttcd3<=clock) {
      formdie=dup;
      sheet3=750; }
/* Check the processing time and buffer*/
if(b3<maxb3 && b2 > 0 && form==up && formdie==dup) {
       if(se3<=clock && b3<=maxb3) {
       b3=b3+1;
       b2=b2-1;
       se3=clock+5.8,sheet3=sheet3-1;} }
```

```
        /*   printf("\n %6.2f %6.2f %6.2f %6.2f
%6.2f",clock,rt,ttr3,ut,ttf3); */
        /*   printf("\n %5d %5d %5d %5d %5d %5d
",b1,b2,b3,sheet1,sheet2,sheet3);*/
          /* Check status of BEND machine*/
        if(bend==up && ttf4<=clock) {
             bend=down;
             r=rand()/32768.;
             rt=15.+10.*r; ttr4=clock+rt; }
        else if(bend==down && ttr4<=clock) {
             bend=up;
             r=rand()/32768.;
             ut=-240.*log(1-r);/*printf("%6.2f",ut);  */
             ttf4=clock+ut; }
         /* Check the status of BEND die*/
         if(bendie==dup && sheet4<=0) {
             bendie=ddown;
             ttcd4=clock+25.; }
        else if(bendie==ddown && ttcd4<=clock) {
             bendie=dup;
             sheet4=600; }
         /* Check the processing time and buffer*/
         if( b3 > 0 && bend==up && bendie==dup) {
              if(se4<=clock) {
              counter=counter+1;
              b3=b3-1;
              se4=clock+6.2; sheet4=sheet4-1;} }
         cb1+=b1;  cb2+=b2;  cb3+=b3;  kont=kont+1;
          clock=clock+delt; }
         /*printf("\n %6.2f %6.2f %6.2f %6.2f
%6.2f",clock,rt,ttr3,ut,ttf3);*/
            printf("\n %5d %5d %5d %5d %5d %5d %5d ",
              b1,b2,b3,sheet1,sheet2,sheet3,sheet4);

            printf("\n clock=%6.1f counter=%d",run, counter);
            printf("\n Av.buffer1=%6.2f Av.buffer2=%6.2f
Av.buffer3=%6.2f",
              cb1/kont,cb2/kont,cb3/kont);
            printf("\n Throughput rate=%6.2f per hour",counter*60./clock);
        }
```

The results obtained are as under :

(*i*) The simulation has been run for 50,000 minutes. The throughput rate of the system, when in-process buffer capacities are limited to 1 – 1 – 1 is 7.67 sheets per hour. When the buffer capacities are raised to 5 – 5 – 5, the throughput rate increases to 7.97 sheets per hour, and remain same when the buffer capacity is further increased to 10 – 10 – 10.

(*ii*) The simulation is carried with buffer capacities of 10 – 10 – 10 and with buffers empty at the start. The average number of sheets held in buffers goes on increasing with the passage of time, and it takes very long time for the system to stabilize. The rise in average stock held, with passage of time is given below.

Clock	*Av. buffer 1*	*Av. buffer 2*	*Av. buffer 3*
500	7.40	0.58	5.11
1000	8.30	2.25	7.26
1500	8.79	4.74	8.07
2000	9.02	5.97	8.48
2500	9.19	6.70	8.73
3000	9.30	7.01	8.88
4000	9.41	7.66	9.13
5000	9.49	7.98	9.25
6000	9.52	7.99	9.27
7000	9.54	8.11	9.34
1000	9.60	8.55	9.45
12000	9.62	8.70	9.48
15000	9.64	8.72	9.51
20000	9.64	8.88	9.57
30000	9.67	8.92	9.59
40000	9.67	8.98	9.60
50000	9.68	9.03	9.62

11.8 Example : Car Wash Station

A self-service car wash station comprises 4 washing stalls and a car waiting area. When a customer is in a stall, he may choose among three options : rinse only; wash and rinse; and wash, rinse and wax. The rinse only takes just 5 minutes, while wash and rinse take 10 minutes, and wash, rinse and wax takes 15 minutes. The past records show that 20% of the customers rinse only while 65% of the customers wash and rinse and 15% of the customers wash, rinse and wax their cars. There are no scheduled appointments. Customers arrive at random, at a rate of about 30 cars per hour. The capacity of the car waiting area is limited to three cars only and currently many customers are lost. The owners of the wash station want to know, how much more business they will do if they add one more stall. Adding a stall will take away one space of the waiting area.

Determine the percentage of lost customer, for the car waiting area of 2, 3 & 4 cars, when the number of working stalls is increased in step of one from 3 to 8.

In the car wash station, there are four washing stalls at present which draw customers from the waiting area, which has a limited capacity. The arrivals to the waiting area are random at a rate of 30 per hour. It can be assumed that the arrivals follow exponential distribution. This is a single queue four servers in parallel queuing model. The service times of the four servers are identical and follow an empirical distribution as under:

Service Time (min.)	*Probability (%)*	*Cumulative Probability (%)*
5	20	20
10	65	85
15	15	100

The simulation of this system can be based on any of the next event increment or the fixed event increment time flow model. The events of interest will be the arrival of the customers, the service completions at various washing stalls, while the state of the system will be described by the state of the queue, state of the stalls being busy or idle. Let us build the model on the fixed increment time flow mechanism. The state of the system at the start of simulation will have significant influence on the results. If we start with empty system, it will take quite long time for the system to reach steady state. It can be assumed that to start with, all the stalls have customers, and the waiting area is empty. As our clock starts, the service on all the four cars starts. The next arrival can take place at any time. The performance of the system is measured by the throughput, that is the number of cars washed which can be represented as percentage of arrivals. Alternatively, it could be the percentage of customers lost. All the three statistics, the customer arrivals, customers served, and customers lost, can be recorded.

The simulation will move through following steps:

1. Initialize the variables according to the assumed starting conditions.
2. The simulation has to continue so long as the clock is less than the prescribed length of run.
3. (At any moment in time) check the state of the queue. If next arrival of car is due advance the arrival counter by one. If there is room in waiting area, add the car to the queue, otherwise add it to the lost.
4. Check the state of each washing stall. If the earlier service has ended, add one to the counter of customers served. If the queue is greater than zero that is cars are available in washing area, draw one, generate service time and fix service ending time.
5. Advance the clock by a fixed time increment.
6. If the clock has reached the predecided length of run, compute the percentage of cars served, percentage of cars lost.

In this case, the idle time of servers or waiting time of customers has not been recorded, as the measure of performance of the system does not require it. The simulation programme can easily be modified to record these statistics.

The C-program of the car wash station simulation is given below:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
    {  /*         CAR WASHING PROBLEM
      arvls -counter of arrivals of cars,
      counter -counter of cars served,
      lost -counter of cars returned without service due to
             lack of parking space,
      qmax -capacity of parking space,
      q -cars waiting in parking at any time,
      k -number of parallel washing stalls,
      se[i] -service ending time of stall i,
      st -service time,
      iat -inter arrival time and nat next arrival time,
      mue -mean arrival rate, run -length of simulation run. */
```

```
int arvls=0,counter=0,lost=0,qmax,q=0,i,k;
float se[10];
float r,st,clock=0.,iat,nat=0.,mue=0.5,run,delt=0.3;
qmax=4;run=5000.;
printf("\n\n Number of washing stalls=");
scanf("%d",&k);
printf("Capacity of waiting space=%d",qmax);
  /* printf("\n clock  queue   lost   nat   se1   se2   se3
se4");*/
r=rand()/32768.;
nat=(-1/mue)*log(1.-r);

for(i=1;i<=k;++i) {

      r=rand()/32768.;
      if(r<=0.2) se[i]=5.;
      else if(r>0.2 && r<=0.85) se[i]=10.;
      else se[i]=15.;    }

while(clock<=run) {
/* Check the state of waiting area*/
if(clock>=nat) {
      if(q<qmax) q=q+1;
      else lost=lost+1;
      r=rand()/32768.;
      iat=(-1/mue)*log(1-r);
      nat=clock+iat;
      arvls=arvls+1; }

/*check the state of washing stalls*/
 for(i=1;i<=k;++i) {
 if(clock>=se[i]){

      if(q>0) {
      q=q-1; r=rand()/32768.;
      if(r<=0.2) st=5.;
      else if(r>0.2 &&r<=0.85) st=10.;
      else st=15.;
      se[i]=clock+st;
      counter=counter+1; } } }

  /*  printf("\n %6.2f %5d %5d %6.2f ",clock,q,lost,nat);
for(i=1;i<=k;++i) {printf("%6.2f",se[i]);}  */

     clock=clock+delt;
     }

 printf("\n Clock=%6.2f Arrivals=%d  Served=%5.2f  Lost=%5.2f",
     clock,arvls,(100.* counter/arvls), (100.*lost/arvls));
}
```

The simulation has been run for 5000 minutes, and the result obtained are as under :

With car waiting area of 3 cars and 4 washing stalls, the customers lost are 20.55% of the arrivals. When the car waiting area is reduced to 2 cars and the number of washing stalls is increased to 5, the percentage of customers lost comes down to 9.88. If the number of servers is maintained at 4, and waiting area is increased to four cars, the customers lost is 17.17.

The percentage of customers lost will obviously come down as the number of washing stalls is increased. The percentage of customers served, customer lost for different capacities of waiting area and number of servers varying from 3 to 8 are given below.

Clock = 5000, Waiting area (q max) = 2

Number of servers (k)	*Arrivals*	*Served %*	*Lost %*
3	2347	62.55	37.37
4	2358	77.06	22.94
5	2338	90.12	9.88
6	2326	95.49	4.47
7	2360	97.46	2.54
8	2448	99.11	0.89

Clock = 5000, Waiting area (q max) = 3

Number of servers (k)	*Arrivals*	*Served %*	*Lost%*
3	2360	63.43	36.48
4	2385	79.41	20.55
5	2364	92.22	7.66
6	2333	97.04	2.96
7	2362	98.90	1.10
8	2428	99.84	0.16

Clock = 5000, Waiting area (q max) = 4

Number of servers (k)	*Arrivals*	*Served %*	*Lost %*
3	2347	62.53	37.37
4	2329	82.83	17.17
5	2316	94.43	5.48
6	2337	98.37	1.69
7	2354	99.58	0.42
8	2449	99.96	0.04

11.9 EXERCISES

1. In a two station manual assembly line, approximately half of the work is done at each of stations A and B. Job first comes to station A, from whereafter doing the necessary work, operator puts it on a

conveyor, which carries it to station B. The capacity of the conveyor is sufficient to allow any length of queue. The distributions of assembly times, as obtained from the past records, are as follows:

Cycle Time (Minutes)	*Frequency of Cycle time* Station A	Station B
0.10	1	0
0.20	3	4
0.30	8	10
0.40	18	16
0.50	24	32
0.60	21	21
0.70	16	12
0.80	6	4
0.90	2	1
1.00	1	0

Assuming that Station A is never starved and Station B is never blocked, and that time of transportation between the stations is negligible, simulate the system for the assembly of 50 parts. Determine the average output rate of the line and the average length of waiting time before Station B.

2. The common maintenance problem with a heavily loaded machine is the failure of its bearings. There are three bearings which cause the trouble and keep the machine down for a considerable part of time. It looks that the present practice of replacing a bearing as and when it fails is not a good policy. It is decided to evaluate the following three alternate policies :

   (*i*) The present policy of replacing the bearing as and when it fails

   (*ii*) Replace all the three bearings when any one of them fails.

   (*iii*) Replace the bearings which have been in use for 1000 or more hours, when a bearing fails.

   It has been observed that it takes 7 hours on the part of one mechanic to replace one bearing. If two bearings are replaced maintenance time is 9 hours, and it takes 11 hours to replace all the three bearings.

   The wages of maintenance mechanic are Rs. 50 per hour, while the machines down time cost Rs. 40 per hour. The cost of each bearing is Rs. 100.

   From the past records the following frequency distribution of the actual working lives of bearings is available.

Bearing life Hours	*Frequency (%)*
600	2
700	6
800	8
900	14
1000	17
1100	20
1200	15
1300	10
1400	7
1500	1

Simulate the system for different replacement policies and identify the best policy. State clearly the assumptions made.

3. Probabilities have been determined for the movement of the ball in a pinball game (Fig. 11.4), as shown below. The points awarded if the ball strikes a given position are also given below. Simulate the paths of five balls and compute the number of points awarded for each ball.

Positions	*Points*
A	400
B	300
C	200
D	100
E	50
F	20

Path	*Probability*	*Path*	*Probability*	*Path*	*Probability*
S to A	0.30	A to A	.25	B to A	.30
B	0.30	B	.25	B	.20
C	0.25	C	.25	C	.20
D	.10	D	.10	D	.15
E	.04	E	.10	E	.10
F	.01	F	.05	F	.05
C to A	.10	D to A	.10	E to A	.15
B	.15	B	.05	B	.10
C	.20	C	.25	C	.15
D	.15	D	.20	D	.15
E	.25	E	.25	E	.20
F	.15	F	.15	F	.25

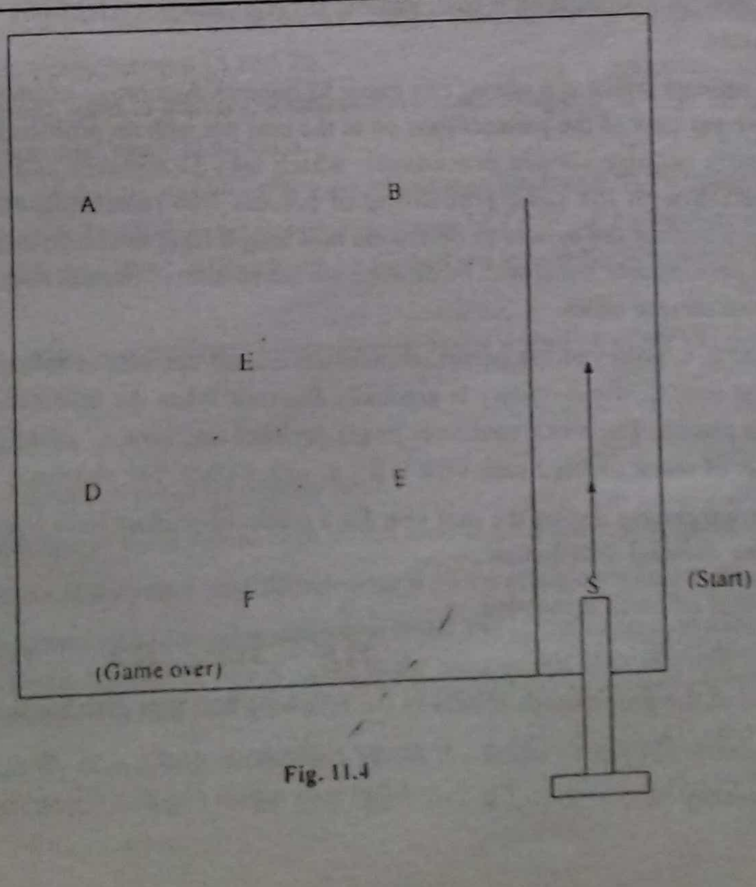

Fig. 11.4

4. In a three station assembly line, there is a very limited space between the stations. Only two assemblies can wait before each of the 2nd and 3rd stations. The first station is never starved, that is, enough jobs are available at the head of the assembly line. Third station is never blocked, that is, enough space is available at the end of the line. The operation times at the three stations are normally distributed with mean 5.0 min, and standard deviation 1.5 min. The production rate of the line is to be determined. Management is also interested to know the increase in production rate, if the space between the stations is increased to allow queues of three assemblies between the stations. Develop a simulation model, run a pilot experiment, determine the simulation run required and then conduct the experiment with inter-station queue capacities of 2, 3 and 4.

5. An elevator in a manufacturing plant carries exactly 500 kg of material. There are three kinds of materials A, B and C and these are in boxes of 200,100, and 50 kg respectively. The inter-arrival times of these boxes for arrival at the elevator door are 5 ± 2 minutes for A, 7.0 minutes constant for B and for C it is 3 minutes in 40% of the cases, and 4 minutes in 60% of the cases. It takes the elevator 1 minute to go up to the first floor, 2 minutes to unload and 1 minute to return to the ground floor. The elevator does not leave the ground floor unless it has a full load. Simulate one hour of operation of the system. What is the average transmit time for a box of material A. Transmit time is the time from the arrival of the box to its unloading. What is the average waiting time for a box of material B? How many boxes of material C made the trip in one hour?

6. The arrival pattern of patients at a clinic for a certain test has been observed to be exponential with mean inter-arrival time of 5 minutes. The test takes exactly 8 minutes and is normally administered by a single doctor. Whenever there are three or more patients in the waiting room, a second doctor also administers the test and continues to do so until the waiting room is empty upon his completing a test. At that point, the second doctor returns to his other duties. If the clinic opens at 9.00 AM, what is the average number of patients in the waiting room from 9.00 to 11.00 AM. For how much time, the second doctor worked in this system. What is the total idle time of the first doctor in the first two hours ?

   Develop a computer simulation model, and run ten replications, each of one day service and find the average values.

7. In a clinic patients arrive at a rate of one every 52 minutes. Audiology examination takes 41 minutes. Seventy-five per cent of the patients pass on to the next test with no problems. Of the remaining 25% patients, 60% require simple procedures, which take 31 minutes and are then sent back for re-examination with the same probability of success. The remaining 40% are sent home with medication. Simulate the system to determine how long it takes to screen and pass 100 patients. It is to note that persons sent home with medication are not considered passed. Assume uniform distribution of arrival and service times.

8. The two bin, S, s, policy of the perpetual inventory system operates as follows. Starting with a stock level of S. at time t_0, the inventory is gradually depleted. When the inventory level reaches s at time t_1 reorder is placed. The stock continues to get depleted until time t_2, when the reorder materialises. The quantity of stock ordered each time is S − s, with s < S/2. The shortages are lost.

   Sales at the warehouse during the past year for a particular product have been analyzed to determine the following demand distribution :

Number of units demanded	0	1	2	3	4
Frequency %	52.5	33.2	10.8	3.0	0.5

   The analysis of the past records results in the following lead time distribution :

Lead time (Days)	2	3	4	5	6
Frequency %	8	18	36	28	10

The inventory cost data is given below :

Carrying cost = Rs. 10 per unit per day

Shortage cost = Rs. 250 per unit short

Reordering cost = Rs. 1200 per reorder

Select your own values of S and s, and run the simulation. Develop a search procedure for reaching optimum values of S and s.

9. The inter-arrival times of people at a self-service cafeteria are uniformly distributed over the range 20 to 80 seconds. Thirty per cent of them go to the sandwich counter, where one worker takes a mean time of 60 seconds with standard deviation of 20 seconds to make a sandwich. The remaining go to the main counter and collect the cooked food, which takes 20 ± 10 seconds. For all customers eating takes 25 ± 10 minutes. After eating, 15% of the customers go for dessert, spending an additional 10 ± 2 minutes. All people make payments at a single counter, which takes 30 ± 5 seconds. Simulate the system to find the average number of customers in the cafeteria. Stop the simulation when 100 people are through the cafeteria. What number of people is left in the cafeteria and what are they doing at the time the simulation stops?

10. A large transport company operates a maintenance and repair department for its fleet of trucks. The time between the arrivals of the trucks is random having exponential distribution, with mean of 20 minutes. Depending upon the service requirements, an arriving truck enters one of the three service areas. Service in all the three areas is on FIFO basis and each service area can handle only one truck at a time. The probabilities of the arrivals entering each of the three service areas are as follows:

Service Area	*Probability*
1	0.45
2	0.30
3	0.25

The service times at the three service areas are random variables having the following distributions :

Area 1: Rectangular between 15 and 25.

Area 2: Normal, mean 15 minutes, standard deviation 5 minutes.

Area 3: Exponential, mean 20 minutes.

After leaving the service area each truck is washed and then lubricated. The washing time is normally distributed with mean 15 minutes and standard deviation of 3 minutes. Lubrication time is a constant 15 minutes for each truck. Only one truck can be washed at a time and only one truck can be lubricated at a time.

Develop a simulation model for the repair system. Draw a flow chart for the model. Design the experiment to determine :

(*i*) Average time a truck spends in the system

(*ii*) Average number of trucks served per hour.

(*iii*) Average length of queue before each of the service areas and before the washing line.

11. Emulate the system of problem 7, and demonstrate it on the graphic terminal of computer.

12. The army headquarters wish to acquire some new tanks. Two different types of tanks, say type X and Y, have been offered. The cost of two X tanks is the same as that of three Y tanks. The operational capabilities of the two tanks are :

(*i*) The probability of an X tank defeating a Y tank in a single shot is 0.50, while that of Y tank defeating X tank in a single shot is 0.4.

(*ii*) If X tank fails to kill a Y tank it fires again at the tank and takes 1 to 3 seconds to fire the next round. If Y tank fails to kill an X tank it fires again within 2 to 4 seconds. Both the times have uniform distributions.

(*iii*) To acquire and engage a new tank the timings are as under :

X tanks 3 ± 2 secs.

Y tanks 6 ± 1 secs.

Both normally distributed.

It has been decided to find out comparative effectiveness of two X tanks against three Y tanks by simulation. The rule to be followed is that both forces attempt to spread fire over opposing tanks as openly as possible.

13. A project comprises a large number of activities, some of which are deterministic in nature, while others are probabilities. The probabilistic activities follow normal distribution for activity durations. The data for a part of the project is given below. The same may be treated as a complete project for the purpose of analysis.

Activity	*Activity time (weeks)*	
	Mean	*S.D.*
1 – 2	10	—
1 – 3	14	1
2 – 3	6	1
2 – 4	12	3
2 – 5	20	2
3 – 4	8	2
4 – 5	12	3

Apply simulation to estimate the probability of completing the project in 36 weeks. Write a computer program and run the simulation on computer for 50, 100 and 500 simulations. Discuss the effect of simulation length on the results. What should be the length of simulation run for the standard error to be within ± 5% of the mean project duration at 95% confidence level?

14. Gill Taxi Service owns a fleet of cars, which are rented to customers. The manager of the taxi stand feels that the number of cars they have at present are inadequate, and that some customers have to be returned without service, and hence there is loss of business. But the addition of a car means an additional cost of Rs. 200 per day. A rented car earns Rs. 400 per day on the average. The past data is available for determining the demand pattern, which is given below:

No of cars demanded per day	*Probability*	*No. of days car rented*	*Probability*
0	.02	1	.30
1	.08	2	.25
2	.25	3	.20
3	.35	4	.15
4	.20	5	.10
5	.08		
6	.02		

Mr. Gill has defined the usage rate of his cars as :

$$\text{Usage rate} = \frac{\text{No. of days cars rented}}{\text{No. of cars with the company} \times \text{days}}$$

The company is interested in knowing the usage rate for different number of cars, that is, if they have 2, 3, ..., 10 cars, and also the optimum number of cars, which may give them a profit of at least Rs. 300 per day.

15. In a machine shop three different machines are available for one hour a day for machining a particular part. Their processing times in seconds are as below :

Machine	*Mean Machining Time*	*Standard Deviation*
1	40	10
2	20	8
3	35	15

Parts arrive by a conveyor at the rate of one every 30 ± 5 seconds for the first three hours of the day. Machine 1 is available for first hour, machine 2 for the second hour and machine 3 for the third hour on each day. Find the number of parts machined per day. What is the largest number of waiting parts? If the parts pile up at any time, discuss why ?

16. A 10 stage flow line production system has normally distributed identical operation times at different stages with mean of 5 minutes and standard deviation of 1.5 minutes. In between the stages, identical spaces are provided for the in-process inventory. There is enough supply at the head of the line so that the first stage is never starved, and there is enough space at the end of the line, so that the last stage is never blocked. The output rate of the line depended upon the amount of in-process inventory held in the interstage spaces.

Develop simulation model of the system to estimate the output rate and the average amount of in-process inventory, when capacity of each interstage buffer is I (I = 1, 2, ..., 10) units. Design the simulation experiment and run the simulation for the designed simulation length.

17. The Research and Development Department of Kalsi Associates has completed the development of their new model of washing machine. Other departments connected with the production of the machine have also completed their studies and have submitted the following reports :

Marketing: The market research team expects sales to be around 3000 units per month with a standard deviation of 400 units, and expect the profits to be around Rs. 120 per unit with probability of 0.6 and Rs. 100 and Rs. 130 with probability of 0.2 each.

Production: The additional plant and machinery and permanent staff will result into total fixed cost of Rs. 10 lacs (50% chances), Rs. 8 lacs (25% chances) and Rs. 12 lacs (25% chances).

Accounts: The accounts department estimates the variable cost to be around Rs. 100 per unit with standard deviation of Rs. 10.

Before starting the production, management wants to be sure of its return and the job has been entrusted to an O.R. analyst.

Develop simulation model for the production system, make 25 runs, design the simulation experiment, and run the simulation to reach some reliable conclusions.

18. Four types of cartons, that is, A, B, C and D, are stored in a warehouse of capacity 2000 cubic meters. Cartons A are of sizes 1 cubic meter, B are of size 2 cubic meters, C are of 3 cubic meters and D are of size 4 cubic meters. The arrival rates of cartons of different types are different. The inter-arrival times of cartons are 10 ± 5 minutes, 10 minutes, 6 ± 6 minutes and 15 ± 10 minutes for carton of type A, B, C and D respectively. If no cartons are removed, how long will it take to fill an empty warehouse, and what will be the number of each type in the warehouse at that time?

19. Rin-Ran Car wash station comprises five stages. Each car takes 31 minutes at each stage. A car can move from one stage to next only when car ahead of it moves, that is, it is a fully coupled system. There is space for 8 cars in the waiting area. Cars arrive at the workstation with a normally distributed inter-arrival time having mean of 6 minutes and standard deviation of 3 minutes. If an arriving car finds the waiting area full it balks away. Determine the balking rate per hour.

20. A salesman of the Cleanwell Vacuum Cleaners moves from house to house to book orders for his product. The number of orders booked at a house depends upon, who answers his door call. The number of units ordered along with the probabilities are as under :

If a man answers the door call,

Order (units)	:	0	1	2	3
Probability	:	.50	.40	.08	.02

If a woman answers the door call,

Order (units)	:	0	1	2	3
Probability	:	.70	.30	00	00

The chance of a man, woman or a child answering the door call are 20%, 60% and 20% respectively.

At each house, he takes about 35 minutes with standard deviation of 15 minutes to demonstrate his device and he works for 8 hours a day. In addition to his pay, company pays him Rs. 20 per unit as sales incentive. If the salesman works for 25 days a month, compute his earnings due to incentive.

Develop a simulation model and run it for sufficient time to obtain results with reasonable accuracy and with reasonable degree of confidence.

21. A company manufactures mopeds. Depending upon the availability of raw materials and other conditions, the daily production has been varying from 196 mopeds to 204 mopeds, whose probability distribution is as given below :

Production per day	:	196	197	198	199	200	201	202	203	204
Probability	:	.05	.09	.12	.14	.20	.15	.11	.08	.06

The finished mopeds are transported in a specially designed three-storeyed Lorry that can accommodate only 200 mopeds. Using following 15 random numbers, 82, 89, 78, 24, 53, 61, 18, 45, 04, 23, 50, 77, 27, 54, 10, simulate the process to find out :

(*i*) What will be the average number of mopeds waiting in the factory?

(*ii*) What will be the average number of empty spaces in the Lorry?

22. A device comprises five components. Their expected lives are normally distributed with mean and standard deviation of 1000 and 200 for elements 1 and 5; 1000 and 50 for elements 2 and 800 and 40 for elements 3 and 4. The system fails when either of 1, 2 and 5 or any one of 3 and 4 fails. Simulate the working of the system for 10,000 hours and determine the number of times the system fails.

Write a computer program for this simulation and run the simulation for a period of 10 lac hours.

23. The computer center of Mechanical Department has two color printers, which are being used by the students, by the faculty members and by one system analyst. The inter-arrival time of students, faculty and system analyst are normally distributed with mean times of 8, 15 and 30 minutes and standard deviations of 3, 5 and 5 minutes respectively. Faculty members have priority over students while the system analyst has priority over all the users. The times spent at the printers by the students, faculty and system analyst are uniformly distributed with 4 ± 2 for students, 5 ± 2 for faculty and 7 ± 3 for system analyst. Irrespective of the priority, no work is interrupted inbetween. Simulate the system for 100 analyst services and determine the number of students and faculty members serviced during this time. Find the utilization of the printers and the average waiting time of users

24. The arrival of jobs at a job shop is random and follow Poisson distribution with overall arrival rate of 1 every 4 hours. The jobs are of four types and each job flows from workstation to workstation in a fixed sequence. The proportion of jobs and their sequences are given in the following table.

Job Type	*Proportion*	*Workstations sequence*
1	0.1	1, 2, 3, 4
2	0.2	1, 3, 4
3	0.3	2, 4, 3
4	0.4	1, 4

The processing times of jobs at all workstations are normally distributed with mean and standard deviation, in hours, as given in the following table.

Type	*Workstation*			
	1	*2*	*3*	*4*
1	(22,4)	(28,5)	(70,5)	(20,3)
2	(20,2)	–	(65,5)	(12,2)
3	–	(22,2)	(50,10)	(10,2)
4	(32,5)	–	–	(14,2)

Assume that stations 1, 2, 3 and 4 have 10, 8, 20 and 8 workers respectively. Each job engages one worker at a station for the duration of processing time. The jobs are processed on the first-in-first-out basis. There is unlimited space for all queues. Simulate the system for 1000 hours, preceded by 200 hours of warming up period. Based on five replications compute a 95% confidence interval for average worker utilisation. Also compute a 95% confidence interval for the mean time spent by a job in the system.