MODELING & SIMULATION |LEVEL 3| LAB 3

Gift Shop Problem

Main Target: Trying to decrease the waiting time of customers in a gift shop.

1. Problem Definition:

Consider a single channel queuing (waiting line) system such as check-out stand at a small gift shop assume that: -

- The time between arrivals of the customers is uniformly distributed form 1 through 10 min (for simplicity round all times to the nearest minute).
- Let's assume that amount of the time required to service each customer is also uniformly distributed form I through 6 minute.
- We are interested in the average time customer spends in the system (both waiting and being serviced) and the percentage of the time that the check-out clerk is not occupied with work.

Simplifying the problem: The whole problem is about decreasing the waiting time of the customer in a gift shop, decreasing the time in which the clerk is idle system

Simulation of Queuing Systems Single server queue:

Queuing system state:

- **System**
  - Server (Clerk)
  - Units (Customer) (in queue or being served)
  - Clock
- **State of the system**
  - Number of units in the system
  - Status of server (idle, busy)
- **Events**
  - Arrival of a unit
  - Departure of a unit

2. Assumption

- **Inter-arrival times:**
  Time between arrivals of customer is uniformly distributed ∈ [1, 10] minute.
- **Service times**
  The Time of service Take ∈ [1, 6] minute
- **Average Waiting Time**
  - Average time that customer spend in a system (waiting time, being served)
  - $A_r = \frac{\sum W_{t(i)}}{n}$

Modern Academy

- **Clerk Idle Time Percentage**
  - The percentage of time that the clerk was Idle or not
  - P= $\sum \frac{Idle_i}{t} * 100$

3. Model

For each customer (i) from customers sample N generate two uniform random numbers

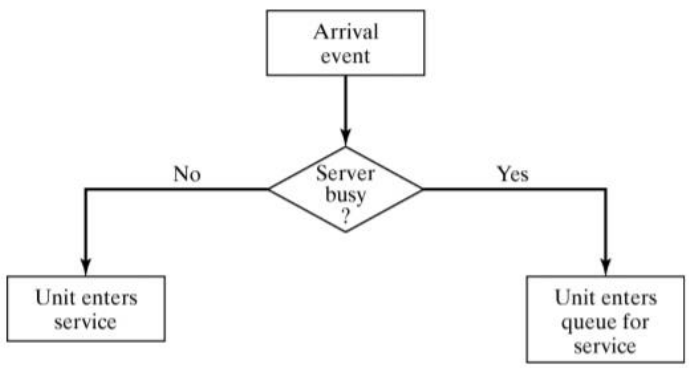
$R_d \in [Low_d, High_d]$ and $R_{st} \in [Low_{st}, High_{st}]$ and compute

1. $Ar\_Clk[i] = Ar\_Clk[i-1] + delay[i]$
2. $Serv\_begin[i] = Max(Ar\_Clk[i]\,, Serv\_end[i-1])$
3. $Serv\_end[i] = Serv\_begin[i] + \; Serv\_time[i]$
4. $WT[i] = Serv.end[i] - Ar\_Clk[i]$
5. $Idle[i] = \begin{cases} Ar_{Clk[i]} - Serv_{end[i-1]} & Ar_{Clk[i]} > Serv\_end[i-1] \\ 0 & otherwise \end{cases}$

4. Flowchart /Algorithm

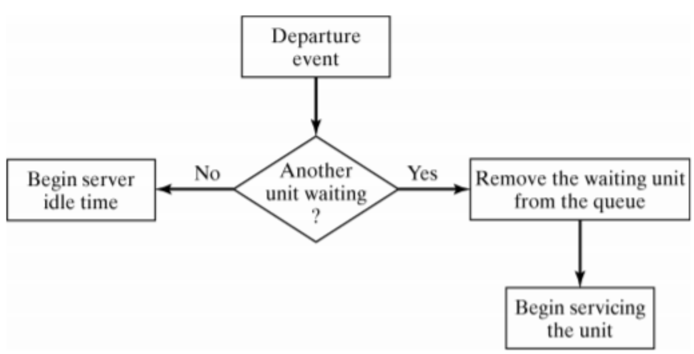
Arrival Event

- If server busy unit enters queue, otherwise unit gets service.



Departure Event

- If queue is not empty begin servicing next unit, otherwise server will be idle.



5. Example:

Customer	Delay	Service time	Arrival Clock	Service begin	Service end	Waiting time	Idle
1	0	2	0	0	2	2	0
2	4	5	4	4	9	5	2
3	2	1	6	9	10	4	0
4	8	4	14	14	18	4	4
5	3	3	17	18	21	4	0

- **Average Waiting Time:**
    - $A_r = \frac{\sum W_{t(i)}}{n} = \frac{19}{5} = 3.8\ min$
- **Clerk Idle Time Percentage:**
    - The percentage of time that the clerk was Idle or not
    - P= $\frac{\sum Idle}{t} * 100 = \frac{6}{21}$ *100 = 28.5%
    - Standard clerk Idle => 12.5 % According to H.R.W

Exercise:

Customer	Delay	Service time	Arrival Clock	Service begin	Service end	Waiting time	Idle
1	0	1	0	0	1	1	0
2	10	4	10	10	14	4	9
3	7	6	17	17	23	6	3
4	7	6	24	24	30	6	1
5	8	2	32	32	34	2	2

- **Average Waiting Time:**
    - $A_r = \frac{\sum W_{t(i)}}{n} = \frac{19}{5} = 3.8\ min$
- **Clerk Idle Time Percentage:**
    - The percentage of time that the clerk was Idle or not
    - P= $\frac{\sum Idle}{t} * 100 = \frac{15}{34}$ *100 = 44.11%
    - Standard clerk Idle => 12.5 % According to H.R.W

Modern Academy

6. Code

```
#include<iostream>
using namespace std;
#define n 20

struct customer
{
	int delay, st, ar_clk, s_begin , s_end , wt, idle ;
};

int RandInt (int Low , int High)
{
	int num;
	double r;
	r= rand() / (double) (RAND_MAX + 1);
	num= (int) (Low+ (r*(High-Low)));
	return num;
}
void main()
{
	customer *c;
	c= new customer[n+1];
	int i, T_time=0 , T_idle=0 ;
	double Av_wait, P;
	cout<< "Cust No."<<"delay"<<"\t st"<<"\t ar_clk"<<"\t s_beg"<<"\t
s_end"<<"\t wait"<<"\t idle"<<endl;

cout<<"====================================================================="<
<endl;
	 // For Customer #1
	i=1;
	c[1].delay=0;
	c[1].st= RandInt(1,6);
	c[1].ar_clk= 0;
	c[1].s_begin= 0;
	c[1].s_end= c[1].st;
	c[1].wt= c[1].st;
	c[1].idle= 0;

	T_time += c[1].wt;

	cout<<i<<"\t"<<c[i].delay<<"\t"<<c[i].st<<"\t"<<c[i].ar_clk<<"\t"<<c[i]
.s_begin<<"\t"<<c[i].s_end<<"\t"<<c[i].wt<<"\t"<<c[i].idle<<endl;
	cout<<"________________________________________________________________
___"<<endl;
```

```
        for (i=2 ; i<=n ; i++)
        {

            c[i].delay= RandInt(1,10);
            c[i].st= RandInt(1,6);
            c[i].ar_clk= c[i-1].ar_clk + c[i].delay; //Ar_Clk[i] = Ar_Clk[i-1] +
delay[i]
            c[i].s_begin= max(c[i].ar_clk , c[i-1].s_end);
                                                //Serv_begin[i]=Max(Ar_Clk[i],
Serv_end[i-1])
            c[i].s_end = c[i].s_begin+ c[i].st; //Serv_end[i] = Serv_begin[i]+
Serv_time[i]
            c[i].wt= c[i].s_end - c[i].ar_clk;  // WT[i] = Serv.end[i] -
Ar_Clk[i]
               if ( c[i-1].s_end < c[i].ar_clk )
                 c[i].idle = c[i].ar_clk - c[i-1].s_end;//Idle[i]={ (Ar_Clk[i]-
Serv_end[i-1]
                                                                     0
otherwise)
               else
                 c[i].idle = 0;

             T_time += c[i].wt;
             T_idle += c[i].idle;

        cout<<i<<"\t"<<c[i].delay<<"\t"<<c[i].st<<"\t"<<c[i].ar_clk<<"\t"<<c[i]
.s_begin<<"\t"<<c[i].s_end<<"\t"<<c[i].wt<<"\t"<<c[i].idle<<endl;

        cout<<"________________________________________________________________
___"<<endl;
        }
         Av_wait = (double) T_time / n;
         P = (double) T_idle / c[n].s_end;
         cout<<endl<<"Average Waiting Time="<<Av_wait<<endl<<endl;
         cout<<"Percentage Clerk idle="<<P*100<<"%"<<"\t\t STANDERD IDLE=12.5%
Accodring H.R.W."<<endl<<endl;

        delete[] c;


}
```

7. Output

```
Cust No.delay    st      ar_clk  s_beg   s_end   wait    idle
=====================================================================
1       0       1       0       0       1       1       0
2       6       1       6       6       7       1       5
3       8       3       14      14      17      3       7
4       5       2       19      19      21      2       2
5       9       5       28      28      33      5       7
6       7       1       35      35      36      1       2
7       8       4       43      43      47      4       7
8       5       2       48      48      50      2       1
9       1       1       49      50      51      2       0
10      4       1       53      53      54      1       2
11      2       5       55      55      60      5       1
12      5       1       60      60      61      1       0
13      1       1       61      61      62      1       0
14      4       3       65      65      68      3       3
15      6       4       71      71      75      4       3
16      6       1       77      77      78      1       2
17      6       3       83      83      86      3       5
18      4       1       87      87      88      1       1
19      6       4       93      93      97      4       5
20      8       3       101     101     104     3       4

Average Waiting Time=2.4
Percentage Clerk idle=54.8077%          STANDERD IDLE=12.5% Accodring H.R.W.
Press any key to continue . . .
```

8. Statistical Analysis

- **Average Waiting Time :**
  - $A_r = \frac{\sum W_{t(i)}}{n} = \frac{48}{20} = 2.4\ min$
  - It is a Good waiting time for customer.
- **Clerk Idle Time Percentage:**
  - The percentage of time that the clerk was Idle or not
  - P= $\frac{\sum Idle}{t} * 100 = \frac{57}{104}$ *100 = 54.8%
  - The Clerk is idle for a long time. Standard clerk Idle => 12.5 % According to H.R.W

Assignment #2:

1- Report about Simulation Language.
2- Simulate this case study (Gift Shop) using excel.

Modern