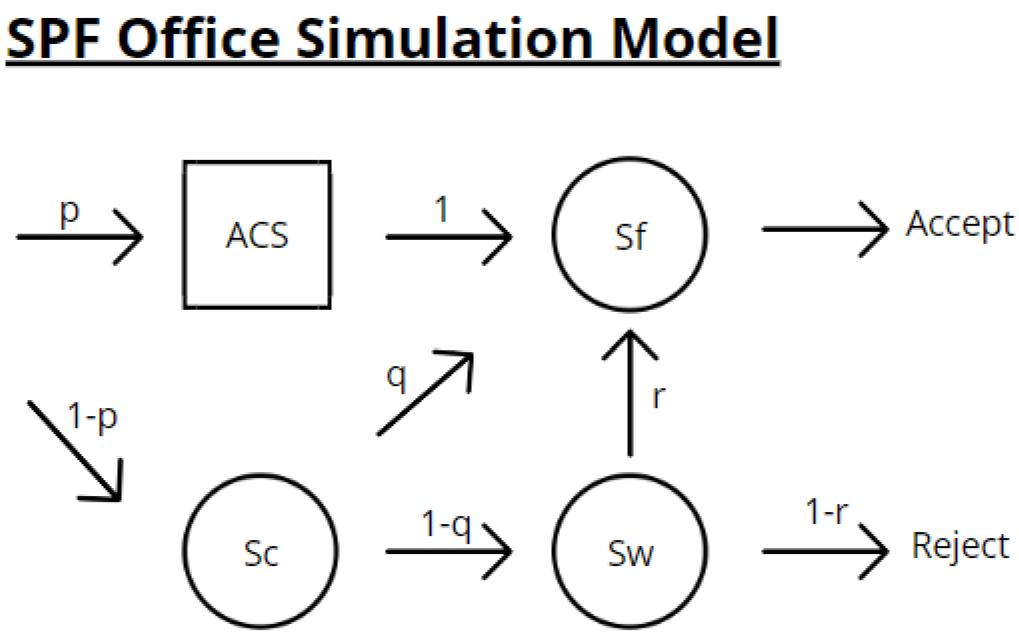
MA424 Modelling in Operations Research : Mock Project Report
Student ID - **202355936**

1) i) Model



PSEUDO CODE - SPF Office Simulation

ST = [[N_A, $n_{SF,}$ $n_{SC,}$ n_{SW}]]
Event_List = [[t_A, t_{SC}, t_{SF}, t_{SW}]]

Initialisation

Time Variable-
t = 0

Events -
t_A = generate_arrival(lambda)
t_{SC} = t_{SF} = t_{SW} = infinity

Counter Variables -
N_{SC} = N_{SF} = N_{SW} = 0 (Number of customers at Server SC, SF and SW)
C_a = C_r = 0 #count of accepted and rejected

$N_{ACS} = 0$

System State Variables -

$n_{SC} = n_{SF} = n_{SW} = 0$

ST = [[N_A, $n_{SF,}$ $n_{SC,}$ n_{SW}]]

ST = [[0, 0, 0, 0]]

Output Variables -

A = []

D = [0]*100000

accepted = []

rejected = []

CASE 1 - (Arrival) min(t_A, t_{SC}, t_{SF}, t_{SW}, T) = t_A

N_A += 1

t= t_A

t_A = generate_arrival(lambda) + t

A(N_A) = t

If runif < p: #completed ACS

    n_{SF} += 1

    N_{SF} += 1

    N_{ACS} +=1

    ST[N_A, SF] = n_{SF} #Update Nath row and SFth column on state matrix

    If n_{SF} ==1:

        t_{SF} = t + generate_service_time(mu$_F$)

else: #no ACS

    n_{SC} += 1

    N_{SC} += 1

    ST[N_A, SC] = n_{SC}

    If n_{SC} ==1:

        t_{SC} = t + generate_service_time(mu$_C$)

Event_list = rbind(event_list, c(t_A, t_{SC}, t_{SF}, t_{SW}))

CASE 2 - (Departure at SC) min(t_A, t_{SC}, t_{SF}, t_{SW}, T) = t_{SC}

t= t_{SC}

n_{SC} -= 1

Person_dept = row_number with ST$SC ==1 #finding out who departed

ST$SC = max(0, ST$SC -1) #updating the queue of SC

If runif < q:
n_{SF} +=1
N_{SF} +=1
ST(person_dept, SF) = max(ST$SF) +1
If n_{sF} ==1:
t_{SF} = generate_service_time(mu_F)
Else:
n_{sw} +=1
N_{SW} +=1
ST(person_dept, SW) = max(ST$SW) +1
If n_{sw} ==1:
t_{SW} = generate_service_time(mu_W)

If n_{SC} == 0:
t_{SC} = infinity
Else:
t_{SC} = generate_service_time(mu_C)

Event_list = rbind(event_list, c(t_A, t_{SC}, t_{SF}, t_{SW}))

CASE 3 - (Departure at SF) min(t_A, t_{SC}, t_{SF}, t_{SW}, T) = t_{SF}

t= t_{SF}
n_{SF} -= 1
Person_dept = row_number with ST$SF ==1 #finding out who departed
ST$SF = max(0, ST$SF -1) #updating the queue of SC
C_A +=1
D[person_dept] = t_{SF}
accepted<- c(accepted, person_dept)

If n_{SF} == 0:
t_{SF} = infinity
Else:
t_{SF} = generate_service_time(mu_F) + t

Event_list = rbind(event_list, c(t_A, t_{SC}, t_{SF}, t_{SW}))
CASE 4 - (Departure at SW) min(t_A, t_{SC}, t_{SF}, t_{SW}, T) = t_{SW}

t= t_{SW}
n_{SW} -= 1
Person_dept = row_number with ST$SW ==1 #finding out who departed
ST$SW = max(0, ST$SW -1) #updating the queue of SC

If runif <=r:

n_{SF} +=1
N_{SF} +=1
ST(person_dept, SF) = max(ST$SF) +1
If n_{sF} ==1:
t_{SF} = generate_service_time(mu_F)

Else: #application rejected, person leaves
C_r += 1
D[person_dept] = t_{SW}
rejected<- c(rejected, person_dept)

If n_{SW} == 0:
t_{SW} = infinity
Else:
t_{SW} = generate_service_time(mu_W) +t

Event_list = rbind(event_list, c(t_A, t_{SC}, t_{SF}, t_{SW}))

#Time ends, stopping conditions begin
CASE 5 - (Departure at SW) min(t_A, t_{SC}, t_{SF}, t_{SW}, T) = T

If t_{SC} = t_{SF} = t_{SW} = infinity:
Tp =max(t-T, 0)
Break #simulation ends
Else if min(t_{SC}, t_{SF}, t_{SW}) == t_{SC}:
Case 2 (Departure at SC) #Repeat Code Block
Else if min(t_{SC}, t_{SF}, t_{SW}) == t_{SF}:
Case 3 (Departure at SF) #Repeat Code Block
Else if min(t_{SC}, t_{SF}, t_{SW}) == t_{SW}:
Case 4 (Departure at SW) #Repeat Code Block

Estimations from Simulation study

$E[T_a]$= 90.9754 minutes
$E[T_c]$= 94.60968 minutes
$E[T_r]$= 61.13089 minutes

ii) We know that the distribution of sample means will follow a normal distribution by the law of large numbers. The normal distribution's mean can be estimated by the sample mean and its standard deviation can be estimated by s/root(n) where n is the sample size of each realization of the distribution.

In our case, this is the number of iterations of our simulation. For the given number of iterations, K=500, we estimate the following for expected time spent by any customer, a customer who has successfully completed his/her application and a customer who has been rejected as the 95% confidence intervals.

$E[T_a]$ = 90.9754 +/- 3.594915 minutes
$E[T_c]$ = 94.60968 +/- 3.931238 minutes
$E[T_r]$ = 61.13089 +/- 2.485681 minutes

This has been calculated using the a z-factor of 1.96 as it is known that 95% of the realizations of a normal distribution lie within 1.96 times the standard deviation away from the mean.

The size of the interval can be adjusted by modifying the number of iterations of our simulation. The 95% confidence interval is calculated using the following formula.

$$\bar{x} \pm 1.96 \left(\frac{s}{\sqrt{n}} \right)$$

where n is the number of iterations of our simulation.

For our true value to lie within 10 minutes away from our estimate, we find out what the value of s/root(n) to be.

s/root(n) = (10/60)/1.96 hours = 0.08503

As s depends on the sample size, we need to iteratively increase the number of iterations till we get the above value. We see following estimates for the different iterations -

Iterations	**Size of confidence interval (One Side) (1.96*s/root(n))*60 (Minutes)**
10	29.18105
20	21.33138
50	12.99975
70	10.52065

80	9.757422

We see that we can estimate the value of E[Ta] for which we are 95% confident that it is within 10 minutes of its true value with 70-80 iterations of our simulation. We get the standard deviation of the estimator with the following command in R.

```
print((sd(Ta_avg_v)/sqrt(K))*60)
```

This comes out to be `4.978277` (For 80 iterations)

iii) For a 90% confidence interval for the same, we first calculate the z-factor for the normal distribution so that 90% of the realizations from the distribution lie in the interval.

```
=NORM.INV(0.95,0,1)
```

We use the above formula in excel to calculate the z-factor for a 90% confidence interval. We use 0.95, as the formula returns inverse of the cumulative distribution function of the normal distribution. A 90% confidence interval corresponds to 0.95 of realizations from the beginning of the left tail of the distribution.

A 90% confidence interval with 80 iterations-
`89.6782984815963 - 106.055371521196 (Minutes)`

iv) We can use counter variables C_a and C_r to keep track of the number of customers who got accepted and rejected. In the code, the vector `percent_comp` is used to store the value of percentage of customers who successfully completed their applications among all customers for all iterations.

Analytically,

Percentage of customers accepted = p + (1-p)*q + (1-p)*(1-q)*r

With
p = 0.5
q = 0.6
r = 0.4

Percentage of customers accepted = 0.5 + (1-0.5)*0.6 + (1-0.5)*(1-0.6)*0.4 = 0.5 +0.3 + 0.08 = **0.88 = 88%**

On simulating with the following number of iterations we get the following values as the average percentage of customers accepted.

Iterations	**mean_percent_comp**
10	0.887034

20	0.8900426
300	0.8792652
500	0.878693

We see that around 300 iterations we see the percentage of accepted customers converging to 88%.
With 300 iterations the 95% confidence interval of the the percentage of accepted customers is `[0.874974465105782, 0.882411527696975]`

Even with a lower number of iterations, we are able to get close to our analytical answer. However, for a more confident estimate, a higher number of iterations is recommended.

We can increase the number of iterations to reduce the size of the confidence interval.

2. Improve the estimator of $E[T_a]$.

Antithetic variables can be used in the simulation to reduce the variance of E[Ta]. We generate a pair of random variables and use them in our simulation. Within each pair, the random numbers are negatively correlated. Therefore, the distribution of the sample estimate we get will have lower standard deviation. This allows us to get a better estimate with a lower number of iterations. A better estimate is characterized by a lower variance in our estimation. This is due to the following formula for variance of the sum of two random variables.

$$\mathrm{Var}((X+Y)/2)=\tfrac{1}{4}(\mathrm{Var}(X)+\mathrm{Var}(Y)+2\mathrm{Cov}(X,Y))$$

If X and Y are negatively correlated then $Cov(X,Y) < 0$, which reduces the variance of the estimator. For 500 iterations the formula becomes the following,

$$\mathrm{Var}((X+Y)/500)=1/500^2(\mathrm{Var}(X)+\mathrm{Var}(Y)+2\mathrm{Cov}(X,Y))$$

In practice, for the first 250 iterations we can use a set of uniform random numbers(U) and for the second 250 iterations we can use (1-U) for the simulation. On calculating the confidence interval of the estimate we will observe a tighter interval for 90% and 95% confidence intervals.