SIMULATIONS MOCK PROJECT 2021-22

Candidate Number 39719

December 13th 2021

INTRODUCTION :

Foodys is a meal delivery service that delivers takeout food from restaurants to clients' homes using a fleet of scooters.The company employs two mechanics, both of whom are qualified to perform inspections and repairs. However, mechanic 1 is specifically assigned for checkups and mechanic 2 for repairs.Foodys is worried about having more than N-n scooters at the mechanics and having to work with less than n scooters as a result. Let T be the first time Foodys works with fewer than n scooters (i.e. they are short-staffed). We want to figure out what E[T] is.

PSEUDO - CODE (Technical):

```
# VARIABLES : time variable t ,
#             Scooters in parking lot -> N-n = 10,
#             number of scooters working -> n = 10,
#             number of scooters in shop -> total_ws_cnt = workshop1_count +
workshop2_count = 0



# EVENT LIST :
# T_A1 = Time of arrival for breakdown repairs at M1
# T_A2 = Time of arrival for checkup at M2
# T_D1 = Time of departure from M1
# T_D2 = Time of departure from M2

# INITIALIZATION OF TIME -

# Set t = 0
# Set N_A1 = N_A2 = N_D1 = N_D2 = 0 (number of scooters arrived and departed
at each mechanic after time t has passed)
# Generate T_A1 and set t = T_A1;
# T_A2, T_D1, T_D2 = Inf
# total_ws_cnt = 0

# CASES -

# Case 0: 1st Breakdown occurs (Generate a time T_A1 for this)
#   set t = T_A1

# Case 1: T_A1 = min{T_A1, T_A2, T_D1, T_D2} (Breakdown)
#   set t = T_A1
#   set N_A1 = N_A1 + 1, -> Since there is an additional arrival at M1
#   set workshop1_count = workshop1_count + 1
#   Generate a checkup time Y2 and set T_A2 = T_D1 + 2
#   if workshop1_count = 1, Generate Y1 (service time of M1) and set T_D1 = t + Y1
#   Check if total_ws_cnt = 11, if yes then store t, if no then find the next minimum

# Case 2: T_D1 = min{T_A1, T_A2, T_D1, T_D2} (Departure from M1)
#   set t = T_D1
#   set N_D1 = N_D1 + 1, -> Since there is an additional arrival
#   set total_ws_cnt = total_ws_cnt - 1
#   if workshop1_count = 0, then set T_D1 = infinity else generate Y1 (service time of M1) and set T_D1 = t+Y1
#   Check if total_ws_cnt = 11, if yes then store t, if no then find the next minimum

# Case 3: T_A2 = min{T_A1, T_A2, T_D1, T_D2} (Check-Up)
#   set t = T_A2
#   set N_A2 = N_A2 + 1, -> Since there is an additional arrival at M1
#   set workshop2_count = workshop2_count + 1
#   Generate Y2 (service time of M2) and set T_D2 = t + Y2
#   Schedule the next checkup time by setting T_A2 = T_D2 + 2
#   Check if total_ws_cnt = 11, if yes then store t, if no then find the next minimum

# Case 4: T_D2 = min{T_A1, T_A2, T_D1, T_D2} (Departure from M2)
#   set t = T_D2
#   set N_D2 = N_D2 + 1, -> Since there is an additional arrival
#   set total_ws_cnt = total_ws_cnt - 1
#   if workshop2_count = 0, then set T_D2 = infinity else generate Y2 (service time of M2) and set T_D2 = t+Y2
#   Check if total_ws_cnt = 11, if yes then store t, if no then find the next minimum
```

IMPLEMENTATION IN R (Technical):

```
# Clear your environment of variables
rm(list = ls())
set.seed(1)
N = 20 #number of scooters owned

n = 10 #number of scooters in service

c = 2 #number of days after which checkup is due

mu = 0.25 #exponential rate at which a scooter in service breaks-down

mu_1 = 1.5 #exponential rate for servicing time of mechanic 1

mu_2 = 2 #exponential rate for for servicing time of mechanic 2

K = 500 #total number of iterations to be done
```

```
source("C:/Users/Arnav Jaitly/Desktop/OR&A/MA
424/Simulation/time_next_arrival.R") #PLEASE EDIT THIS PART FOR THE CODE TO
RUN

t = 0
N_A1 = 0
N_A2 = 0
N_D1 = 0
N_D2 = 0
workshop1_count = 0
workshop2_count = 0

t_A2 = Inf
t_D1 = Inf
t_D2 = Inf
t_A1 = -(1/mu) * log(runif(1)) # Case 0, generating a breakdown time

ST1 = matrix(c(workshop1_count,t) , nrow = 1, ncol = 2)
ST2 = matrix(c(workshop2_count,t) , nrow = 1, ncol = 2)

event_list1 =  matrix(c(t_A1, t_D1), nrow=1, ncol=2)
event_list2 =  matrix(c(t_A2, t_D2), nrow=1, ncol=2)

lambda = 7

t_column = matrix(c(t),nrow =1, ncol =1)

for(i in 1:500){
  t = 0
  N_A1 = 0
  N_A2 = 0
  N_D1 = 0
  N_D2 = 0
  workshop1_count = 0
  workshop2_count = 0

  t_A2 = Inf
  t_D1 = Inf
  t_D2 = Inf
  t_A1 = -(1/mu) * log(runif(1)) # Case 0, generating a breakdown time

  flag = 1
  while(flag){

    if (((t_A1 <= t_D1) & (t_A1 <= t_D2) & (t_A1 <= t_A2)) & (workshop1_count
+ workshop2_count < 11)){ # Case 1:

      t = t_A1 # move to time t_A1
      N_A1 = N_A1 + 1 # count the arrival
      workshop1_count = workshop1_count + 1 # count the state
      t_A1 = time_next_arrival(t, lambda) # generate time of next arrival
      if (workshop1_count == 1){ # system had been empty so the arrival goes
to the mechanic
        Y1 = -(1/mu_1) * log(runif(1)) # generate an exp(mu_1) RV
        t_D1 = t + Y1 # set time of next departure
      }
      A1 = c(N_A1,t) # collect output data: A(N_A) = t
      ST1 = rbind(ST1, c(workshop1_count,t)) # update ST
      event_list1 = rbind(event_list1, c(t_A1,t_D1))


    } else if (((t_D1 < t_A1) & (t_D1 <= t_A2) & (t_D1 <= t_D2)) &
(workshop1_count + workshop2_count < 11)){ # Case 2: departure from Mechanic
1

      t = t_D1 # move to time t_D
      N_D1 = N_D1 + 1 # count the departure
      workshop1_count = workshop1_count - 1 # one less scooter in the system
      if (workshop1_count == 0){ # no customers at the server
        t_D1 = Inf
      } else { # new scooter at the server
        Y1 = -(1/mu_1) * log(runif(1)) # generate an exp(mu_1) RV
        t_D1 = t + Y1 # set time of next departure
      }
      D1 = c(N_D1,t) # collect output data: D(N_D) = t
      ST1 = rbind(ST1, c(workshop1_count,t)) # update ST
      event_list1 = rbind(event_list1, c(t_A1,t_D1))

    } else if (((t_A2 <= t_D1) & (t_A2 <= t_D2) & (t_A2 <= t_A1))&
(workshop1_count + workshop2_count < 11)){
      # Case 3: time ended, customers remain, go to next departure

      t = t_A2
      N_A2 = N_A2 + 1 # count the arrival at the 2nd Mechanic
      workshop2_count = workshop2_count + 1 # count the state
      t_A2 = time_next_arrival(t, lambda) # generate time of next arrival

      if (workshop2_count == 1){ # system had been empty so the arrival goes
to the mechanic
        Y2 = -(1/mu_2) * log(runif(1)) # generate an exp(mu_1) RV
        t_D2 = t + Y2 # set time of next departure
      }
      A = c(N_A2,t) # collect output data: A(N_A) = t
      ST2 = rbind(ST2, c(workshop2_count,t)) # update ST
      event_list2 = rbind(event_list2, c(t_A2,t_D2))


    } else if (((t_D2 < t_A1) & (t_D2 <= t_A2) & (t_D2 <= t_D1)) &
(workshop1_count + workshop2_count < 11)){ #Case 4:
      t = t_D2 # move to time t_D
      N_D2 = N_D2 + 1 # count the departure
      workshop2_count = workshop2_count - 1 # one less scooter in the system
      if (workshop2_count == 0){ # no customers at the server
```

```
        t_D2 = Inf
      } else { # new scooter at the server
        Y2 = -(1/mu_2) * log(runif(1)) # generate an exp(mu_1) RV
        t_D2 = t + Y2 # set time of next departure
      }
      D2 = c(N_D2,t) # collect output data: D(N_D) = t
     ST2 = rbind(ST2, c(workshop2_count,t)) # update ST
     event_list2 = rbind(event_list2, c(t_A2,t_D2))

    } else if(workshop1_count + workshop2_count >= 11) {
      flag = 0
     t_column = rbind(t_column, t)
    }
  }

}
```

```
print("The estimate E[T]: ")
```

```
## [1] "The estimate E[T]: "
```

```
print(mean(t_column))
```

```
## [1] 8.259773
```

COMMENTS FOR CEO:

The estimate number of days after which Foodys can expect to be under - staffed is 8.259773 days. The estimation is based on service efficiency of both the newly recruited mechanics. The assumption here is that the mechanics work without break and continually.