DES Priority Setting

Alireza Sheikh-Zadeh, Ph.D.

In a queuing system, some customers compete for resources by moving through processes. A priority discipline allows the customers to be ordered within the queue by a specified priority or characteristic. For example, the waiting items may be arranged by the due date for a customer order or the severity of the patients' health condition in a walk-in clinic.

In order to consider priorities for our entities, we add these elements to the original model:

- branch()
- set_prioritization()
- set_attribute()

Example: A walk-in health care clinic has analyzed their operations. They have found that they can classify their walk-in patients into two categories: high priority (urgent need of medical attention) and low priority. On a typical day during the period of interest, 50% of patients are high priority, and the remaining are low priority. Other assumptions are the same as the original assumptions in the first Simmer question.

- The clinic opens from 8 am to 5 pm (540 minutes)
- Nurse service time is normally distributed: Normal(mean = 15, sd = 1) minutes
- Doctor service time is normally distributed: Normal(mean = 20, sd = 1) minutes
- Administrator service time is normally distributed: Normal(mean = 5, sd = 1) minutes
- Patients inter-arrival time is normally distributed: Normal(mean = 5, sd = 0.5) minutes

```
library(simmer)

# Set up a trajectory for a patient
###### The Main Trajectory ###################################
patient <- trajectory("patients' path") %>%

  branch(function() sample(1:2, size=1, prob=c(0.50,0.50),replace=TRUE),  con
tinue=c(T,T),

  ###### sub trajectory A ###################################
  trajectory("A") %>%
  set_prioritization(values = c(5,7,T)) %>%

  seize("nurse", 1) %>%
  timeout(function() rnorm(1, 15, 1)) %>%
  release("nurse", 1) %>%
```

```
  seize("doctor", 1) %>%
  timeout(function() rnorm(1, 20, 1)) %>%
  release("doctor", 1) %>%

  seize("administration", 1) %>%
  timeout(function() rnorm(1, 5, 1)) %>%
  release("administration", 1) %>%

  set_attribute("Priority", 1),

  ###### sub trajectory B ###################################
  trajectory("B") %>% # a sub trajectory
  set_prioritization(values = c(3,7,T)) %>%

  seize("nurse", 1) %>%
  timeout(function() rnorm(1, 15, 1)) %>%
  release("nurse", 1) %>%

  seize("doctor", 1) %>%
  timeout(function() rnorm(1, 20, 1)) %>%
  release("doctor", 1) %>%

  seize("administration", 1) %>%
  timeout(function() rnorm(1, 5, 1)) %>%
  release("administration", 1) %>%

  set_attribute("Priority", 2)
  )

#library(simmer.plot)
#plot(patient, verbose = T)

set.seed(123)

envs <- lapply(1:30, function(i) {
  env <- simmer("Clinic") %>%
  add_resource("nurse", 2) %>%
  add_resource("doctor", 3) %>%
  add_resource("administration", 2) %>%
  add_generator("patient", patient, function() rnorm(1, 5, 0.5), mon = 2) %>%
  run(540)
})

patientAttr <- get_mon_attributes(envs)
#head(patientAttr)
# Count of discharged patients for both types in 30 replications
table(patientAttr$value)
```

```
##
##    1    2
## 1460  534
```

```
# Average of discharged patients per replication
colMeans(table(patientAttr$replication, patientAttr$value))
```

```
##        1        2
## 48.66667 17.80000
```

```
x1 <- get_mon_arrivals(envs)
x2 <- get_mon_attributes(envs)

all <- merge(x1, x2, by=c("name", "replication"), all = T)

head(all)
```

```
##       name replication start_time end_time activity_time finished     time
## 1 patient0           1   4.719762 45.24844      40.52868     TRUE 45.24844
## 2 patient0          10   5.015139 45.97090      40.95576     TRUE 45.97090
## 3 patient0          11   5.658330 46.99411      41.33578     TRUE 46.99411
## 4 patient0          12   5.677462 47.64740      41.96994     TRUE 47.64740
## 5 patient0          13   3.962572 44.77832      40.81575     TRUE 44.77832
## 6 patient0          14   5.562715 43.39526      37.83255     TRUE 43.39526
##        key value
## 1 Priority     2
## 2 Priority     1
## 3 Priority     2
## 4 Priority     1
## 5 Priority     2
## 6 Priority     1
```

```
priori1 <- na.omit(subset(all, all$value == 1))
priori2 <- na.omit(subset(all, all$value == 2))

priori1.waiting =  (priori1$end_time-priori1$start_time) - priori1$activity_t
ime
priori2.waiting =  (priori2$end_time-priori2$start_time) - priori2$activity_t
ime

# Overal average waiting time for each type
mean(priori1.waiting)
```

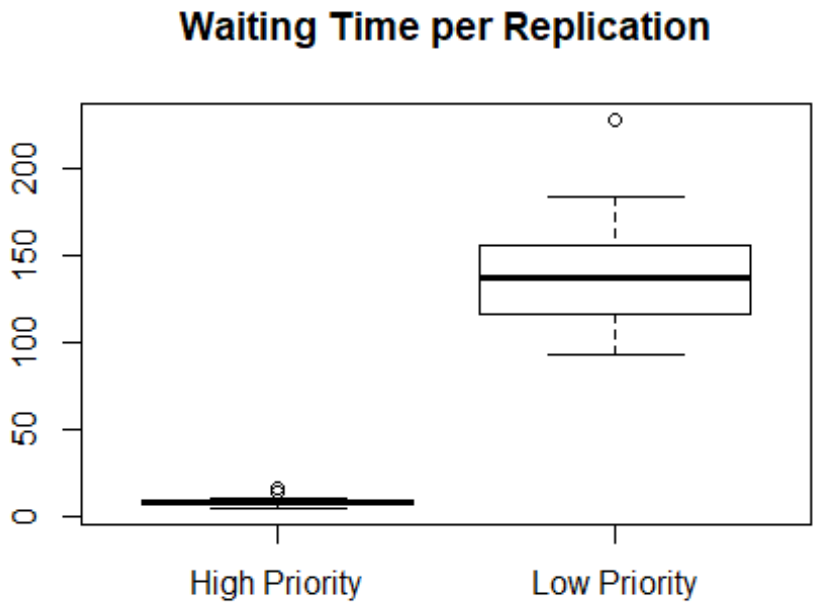
```
## [1] 8.729509
```

```
mean(priori2.waiting)
```

```
## [1] 137.4483
```

```
# Average waiting time per replication for each type
priori1.waiting.rep <- aggregate(priori1.waiting, by = list(priori1$replicati
```

```
on), mean)
priori2.waiting.rep <- aggregate(priori2.waiting, by = list(priori2$replicati
on), mean)
boxplot(priori1.waiting.rep$x, priori2.waiting.rep$x, names = c("High Priorit
y", "Low Priority"), main = "Waiting Time per Replication")
```




```
# Practice: Find the standard error and 95% confidence interval for average w
aiting time for each type of patient.
```

Practice

A clinic has four doctors on staff and one nurse to serve patients during the period of interest. On a typical day during the period of interest, there are about 15 arrivals per hour (therefore, the time between arrivals is exponential with the mean of 60/15 minutes), with 25% being a high priority, 60% being a medium priority, and the remaining being low priority. Upon arrival at the clinic, the patients are triaged by a nurse into one of the three types of patients. This takes only 2 to 3 minutes uniformly distributed. Then, the patients wait in the waiting room and get called to visit doctors on an FCFS basis. The service time distributions of the patients are given as follows.

Priority	Service Time Distribution (in Minutes)
High	Normal(38, 8)
Medium	Triangular(16, 22, 28)
Low	Normal(12, 2)

The clinic would like to estimate the following:

(a) the average flow time of each type of patient,

(b) the 95% confidence interval for the average flow time of each type of patient.