

Proyectos de Simulación basada en Eventos Discretos

Annie Hernández Pérez

1. La Cocina de Kojo (Kojo's Kitchen)

La cocina de *Kojo* es uno de los puestos de comida rápida en un centro comercial. El centro comercial está abierto entre las 10 : 00 am y las 9 : 00 pm cada día. En este lugar se sirven dos tipos de productos: sándwiches y sushi. Para los objetivos de este proyecto se asumirá que existen solo dos tipos de consumidores: unos consumen solo sándwiches y los otros consumen solo productos de la gama del sushi. En Kojo hay dos períodos de hora pico durante un día de trabajo; uno entre las 11 : 30 am y la 1 : 30 pm, y el otro entre las 5 : 00 pm y las 7 : 00 pm. El intervalo de tiempo entre el arribo de un consumidor y el de otro no es homogéneo pero, por conveniencia, se asumirá que es homogéneo. El intervalo de tiempo de los segmentos homogéneos, distribuye de forma exponencial. Actualmente dos empleados trabajan todo el día preparando sándwiches y sushi para los consumidores. El tiempo de preparación depende del producto en cuestión. Estos distribuyen de forma uniforme, en un rango de 3 a 5 minutos para la preparación de sándwiches y entre 5 y 8 minutos para la preparación de sushi.

El administrador de *Kojo* está muy feliz con el negocio, pero ha estado recibiendo quejas de los consumidores por la demora de sus peticiones. Él está interesado en explorar algunas opciones de distribución del personal para reducir el número de quejas. Su interés está centrado en comparar la situación actual con una opción alternativa donde se emplea un tercer empleado durante los períodos más ocupados. La medida del desempeño de estas opciones estará dada por el porcentaje de consumidores que espera más de 5 minutos por un servicio durante el curso de un día de trabajo. Se desea obtener el porcentaje de consumidores que esperan más de 5 minutos cuando solo dos empleados están trabajando y este mismo dato agregando un empleado en las horas pico.

2. Solución

La solución al problema consiste en una implementación en *Python* de un programa que solo recibe si se quiere realizar la simulación utilizando un tercer empleado o no; e imprime un log con todos los eventos que van ocurriendo (llega un cliente, un empleado atiende a un cliente y se marcha un cliente), además del estado de la cola (una lista donde las 3 primeras posiciones son los índices de los clientes que están siendo atendidos por cada uno de los empleados, -1 si no hay ninguno, y las siguientes son los índices de los clientes que están esperando en cola) y al final de cada simulación el porcentaje de los clientes que tuvieron

que esperar más de 5 minutos en la cola.

En el programa luego de la inicialización se va a estar ejecutando un ciclo en el que se va a estar determinando cuál es el próximo evento que va a ocurrir; estos eventos son:

- El nuevo empleado comienza a trabajar: Si estamos usando un tercer empleado, este está libre, hay clientes esperando y es tiempo de que comience a trabajar este va a atender al primer cliente en la cola.
- Llegada de un cliente: Si el próximo evento es la llegada de un cliente y hay algún empleado disponible este lo atiende, sino se pone al final de la cola.
- Salida del cliente atendido por el empleado i ($i = \{1, 2, 3\}$): Si el próximo evento es la salida del cliente que está siendo atendido por uno de los empleados entonces se ve si ya no va hay más clientes por atender y pasó la hora de cerrar para terminar la ejecución, si no hay más clientes en la cola pero no es hora de cerrar entonces se pone al empleado como libre y si quedan clientes esperando se hace que este atienda al próximo, a menos que este sea el tercer empleado y haya terminado su horario de trabajo.

3. Modelo de Simulación

3.1. Variables:

- t : tiempo transcurrido.
- na : cantidad de clientes que han llegado.
- ne : si se va a usar un tercer empleado en la simulación (se recibe como parámetro).
- $(n, i_1, i_2, i_3, \dots, i_k)$: estado del puesto, el primer número es la cantidad de clientes que hay, el segundo el índice del cliente que está siendo atendido por el empleado 1 (-1 si no hay ninguno), el tercero el del que está en el empleado 2 y el cuarto el del empleado 3 (si este se está utilizando, sino va a ser siempre -1), luego habrá $n - 3$ índices correspondientes a los clientes que se encuentran en cola, esperando para ser atendidos. Se va a poner $(-)$ en una de las posiciones de los empleados cuando este puede estar libre o atendiendo un cliente y cualquiera sea su estado este no se va a modificar en el evento actual. Cuando está vacío solo se pone (0) que es que no hay ningún cliente, ni en cola ni con un empleado.
- t_A : tiempo del próximo arribo de un cliente ($t_A \sim T$)
- t_1 : tiempo de salida para el cliente que está siendo atendido por el empleado 1
- t_2 : tiempo de salida para el cliente que está siendo atendido por el empleado 2

- t_3 : tiempo de salida para el cliente que está siendo atendido por el empleado 3(sino se está usando es siempre ∞)
- ct : minuto al que cierra el puesto, tomando el minuto que abre como el 0.
- $brh1$: minuto en el que empieza el primer momento de hora pico.
- $erh1$: minuto en el que termina el primer momento de hora pico.
- $brh2$: minuto en el que empieza el segundo momento de hora pico.
- $erh2$: minuto en el que termina el segundo momento de hora pico.
- $order$: para determinar si un cliente va a pedir sushi o sandwich ($order \sim O$)
- $sushi$: tiempo en que va a estar lista una orden de sushi. ($sushi \sim Su$)
- $sandwich$: tiempo en que va a estar lista una orden de sandwich. ($sandwich \sim Sa$)

Salida:

- A : array donde en la posición i va a estar el tiempo de arribo del i -ésimo cliente.
- D : array donde en la posición i va a estar el tiempo en que fue atendido i -ésimo cliente.

3.2. Simulación:

- Inicialización:
 - $t = na = 0$
 - $SS = (0)$
 - //Generar el tiempo de llegada del primer cliente
 - Generar T
 - $t_A = T$
 - $t_1 = t_2 = t_3 = \infty$
 - $ct = 660$
 - $brh1 = 90$
 - $erh1 = 210$
 - $brh2 = 420$
 - $erh2 = 540$
- //Si hay clientes esperando y estamos usando un tercer camarero y este está libre y le toca empezar a trabajar este debe atender al próximo cliente
- Caso 0: $SS = (n, i_1, i_2, -1, i_3 \dots) \wedge ne \wedge ((\min(t_A, t_1, t_2) \geq brh1 \wedge t \leq erh1) \vee (\min(t_A, t_1, t_2) \geq brh2 \wedge t \leq erh2))$
 - //Actualizar el tiempo según el intervalo en que nos encontremos
 - if $t \leq erh1$:
 - $t = brh1$
 - endif

```

else:
     $t = brh2$ 
endelse
 $D[i_3] = t$ 
//Hacer que el camarero atienda al 1er cliente en la cola
 $SS = (n, i_1, i_2, i_3, \dots)$ 
Generar  $O$ 
//Si el cliente pide sushi
if  $O$ :
    Generar  $Su$ 
     $t_3 = t + Su$ 
endif
//Si el cliente pide sandwich
else:
    Generar  $Sa$ 
     $t_3 = t + Sa$ 
endelse

```

//El próximo evento es la llegada de un cliente

- Caso 1 : $t_A = \min(t_A, t_1, t_2, t_3) \wedge t_A \leq ct$:


```

 $t = t_A$ 
 $na = na + 1$ 
 $A[na] = t$ 
//Generar el tiempo de llegada del próximo cliente
Generar  $T$ 
 $t_A = t + T$ 
//Si el sistema estaba vacío y llegó un cliente, hacer que lo atienda el
empleado 1 y generar su tiempo de salida
if  $SS = (0)$ :
     $D[na] = t_A$ 
     $SS = (1, na, -1, -1)$ 
    Generar  $O$ 
    //Si el cliente pide sushi
    if  $O$ :
        Generar  $Su$ 
         $t_1 = t + Su$ 
    endif
    //Si el cliente pide sandwich
    else:
        Generar  $Sa$ 
         $t_1 = t + Sa$ 
    endelse
endif
//Si solo había un cliente y lo estaba atendiendo el empleado 1, hacer que
al nuevo cliente lo atienda el empleado 2 y generar su tiempo de salida
else if  $SS = (1, j, -1, -1)$ :
     $D[na] = t_A$ 
     $SS = (2, j, na, -1)$ 

```

```

Generar  $O$ 
//Si el cliente pide sushi
if  $O$ :
Generar  $Su$ 
 $t_2 = t + Su$ 
endif
//Si el cliente pide sandwich
else:
Generar  $Sa$ 
 $t_2 = t + Sa$ 
endelse
endif
//Si solo había un cliente y lo estaba atendiendo el empleado 2, hacer que
al nuevo cliente lo atienda el empleado 1 y generar su tiempo de salida
else if  $SS = (1, -1, j, -1)$ :
 $D[na] = t_A$ 
 $SS = (2, na, j, -1)$ 
Generar  $O$ 
//Si el cliente pide sushi
if  $O$ :
Generar  $Su$ 
 $t_1 = t + Su$ 
endif
//Si el cliente pide sandwich
else:
Generar  $Sa$ 
 $t_1 = t + Sa$ 
endelse
endif
//Si estamos usando al 3er camarero y es una hora en la que trabaja y
está disponible entonces atiende al cliente
else if  $SS = (2, i, j, -1) \wedge ne \wedge ((t_A \geq brh1 \wedge t_A \leq erh1) \vee (t_A \geq brh2 \wedge t_A \leq erh2))$ :
 $D[na] = t_A$ 
 $SS = (3, i, j, na)$ 
Generar  $O$ 
//Si el cliente pide sushi
if  $O$ :
Generar  $Su$ 
 $t_3 = t + Su$ 
endif
//Si el cliente pide sandwich
else:
Generar  $Sa$ 
 $t_3 = t + Sa$ 
endelse
endif //Si ningún camarero puede atenderlo poner al cliente al final de la
cola else:
 $SS = (n + 1, i_1, i_2, -, \dots, na)$ 
endelse

```

//El próximo evento es la salida del cliente atendido por el empleado 1

■ Caso 2 : $SS = (n, i_1, -, -, \dots) \wedge t_1 < t_A \wedge t_1 \leq t_2 \wedge t_1 \leq t_3$:

$t = t_1$
//Si solo había un cliente ahora el sistema está vacío y no puede haber una próxima salida
if $n = 1$:
 $SS = (0)$
 $t_1 = \infty$
endif
//Si solo había dos clientes el otro lo está atendiendo otro empleado, el empleado 1 no tiene más ningún cliente por atender
else if $n = 2$:
 $SS = (1, -1, -, -)$
 $t_1 = \infty$
endif
//Si había 3 clientes y cada uno está siendo atendido por un empleado, el empleado 1 no tiene más ningún cliente por atender
else if $SS = (3, i_1, i_2, i_3)$:
 $SS = (2, -1, i_2, i_3)$
 $t_1 = \infty$
endif
//Hacer que el empleado 1 atienda al próximo cliente y generar su tiempo de salida
else:
 $SS = (n - 1, i_4, i_2, \dots, i_n)$
 $D[i_4] = t$
Generar O
//Si el cliente pide sushi
if O :
Generar Su
 $t_1 = t + Su$
endif
//Si el cliente pide sandwich
else:
Generar Sa
 $t_1 = t + Sa$
endelse
endelse

//El próximo evento es la salida del cliente atendido por el empleado 2

■ Caso 3 : $SS = (n, -, i_2, -, \dots) \wedge t_2 < t_A \wedge t_2 < t_1 \wedge t_2 \leq t_3$:

$t = t_2$
//Si solo había un cliente ahora el sistema está vacío y no puede haber una próxima salida
if $n = 1$:

```

SS = (0)
t2 = ∞
endif
//Si solo había dos clientes el otro lo está atendiendo otro empleado, el
empleado 2 no tiene más ningún cliente por atender
else if n = 2:
SS = (1, -, -1, -)
t2 = ∞
endif
//Si había 3 clientes y cada uno está siendo atendido por un empleado, el
empleado 2 no tiene más ningún cliente por atender
else if(SS = (3, i1, i2, i3)):
SS = (2, i1, -1, i3)
t2 = ∞
endif
//Hacer que el empleado 1 atienda al próximo cliente y generar su tiempo
de salida
else:
D[i4] = t
SS = (n - 1, i1, i4, i3, ...)
Generar O
//Si el cliente pide sushi
if O:
Generar Su
t2 = t + Su
endif
//Si el cliente pide sandwich
else:
Generar Sa
t2 = t + Sa
endelse
endelse

```

//El próximo evento es la salida del cliente atendido por el empleado 3

- Caso 4 : $SS = (n, -, -, i_3, \dots) \wedge t_3 < t_A \wedge t_3 < t_1 \wedge t_3 < t_2$:


```

t = t3
//Si solo había un cliente ahora el sistema está vacío y no puede haber
una próxima salida
if n = 1:
SS = (0)
t3 = ∞
endif
//Si solo había dos clientes el otro lo está atendiendo otro empleado, el
empleado 3 no tiene más ningún cliente por atender
else if n = 2:
SS = (1, -, -, -1)
t3 = ∞
endif

```

```

//Si había 3 clientes y cada uno está siendo atendido por un empleado, el
empleado 3 no tiene más ningún cliente por atender
else if  $SS = (3, i_1, i_2, i_3)$ :
 $SS = (2, i_1, i_2, -1)$ 
 $t_3 = \infty$ 
endif
//Si hay más clientes por atender y todavía es su horario de trabajo,
hacemos que atienda al próximo cliente
else if  $(t \geq brh1 \wedge t \leq erh1) \vee (t \geq brh2 \wedge t \leq erh2) \wedge n > 3$ :
 $D[i_4] = t$ 
 $SS = (n - 1, i_1, i_2, i_4, \dots)$ 
Generar  $O$ 
//Si el cliente pide sushi
if  $O$ :
Generar  $Su$ 
 $t_3 = t + Su$ 
endif
//Si el cliente pide sandwich
else:
Generar  $Sa$ 
 $t_3 = t + Sa$ 
endelse
endif
else:
 $SS = (n - 1, -, -, -1, \dots)$ 
 $t_3 = \infty$ 

```

4. Resultados

Si como promedio llegan 8 clientes en cada hora durante el horario normal y 14 clientes en cada hora durante el horario pico y solamente usamos dos empleados, entonces como promedio entre el 9 % y el 11 % de estos van a tener que esperar más de 5 minutos en la cola para ser atendido. Si bajo estas mismas circunstancias contratamos al tercer empleado durante el horario pico, entonces solo entre el 1 % y el 3 % de los clientes van a esperar más de 5 minutos por ser atendidos, por lo que vemos que la situación mejora con el nuevo empleado. Si la cantidad de clientes que llegan en una hora es menor que esta, la incorporación del nuevo empleado no es una mejora significativa. Si esta cantidad es mayor pero la diferencia entre los que llegan en horario normal y en horario pico se mantiene entonces la mejora del nuevo empleado va a ser la misma; si aumenta la diferencia entre los de clientes que llegan en horario pico y los que llegan en horario normal entonces la mejora introducida por el nuevo empleado es mayor, lo contrario también se cumple.