Universidad de La Habana
Facultad de Matemática y Computación

PROYECTO: Simulación de eventos discretos

Emrys Cruz Viera
C311
Angel Daniel Alonso Guevara
C311
Bruno Jesús Pire Ricardo
C311

ÍNDICE

1. INTRODUCCIÓN

Este proyecto tiene como objetivo principal adquirir conocimientos sobre la simulación de eventos discretos y aplicar los conocimientos y habilidades adquiridas en los cursos de estadísticas y probabilidades. Como medio utilizaremos un problema del campo de la **Teoría de colas**.

1.1. Objetivos y metas

- Implementar una simulación que modele el problema planteado a continuación.
- Computar con la mayor precisión posible las variables de interés.
- Comprobar si nuestra propuesta de gestión de colas es mejor que la actual del negocio.

1.2. Sistema específico a simular y las variables de interés

El restaurante chino "Gran Muralla" sirve dos tipos de platos para llevar, los rollitos de primavera y el pollo frito. Hay dos ventanas separadas, una para cada menú. Los clientes llegan según una distribución de Poisson de media 20/hora. El 60 % va a por rollitos y el resto a por el pollo. El 20 % de los que van a por los rollitos pasan luego a por pollo y el resto abandona el restaurante. El 10 % de los que primero se han puesto en la ventana del pollo pasan luego a por rollitos de primavera y el resto abandona el restaurante. Se tarda 4 minutos en servir los rollitos y 5 minutos en servir el pollo frito, el tiempo de servicio es exponencial. ¿Cuánta gente hay por término medio en el restaurante? ¿Cuál es el tiempo medio de espera en cada ventanilla? Si alguien quiere los dos menús ¿Cuánto tiempo pasa en el restaurante? (SABATER, 2015/2016)

De la definición del problema obtenemos que las variables de interes son:

- $\bar{L}$: promedio de personas en el restaurante
- $\bar{W_P}$: promedio del tiempo de espera en la cola del pollo frito
- $\bar{W_R}$: promedio del tiempo de espera en la cola de los rollitos
- $\bar{W_{PR}}$: promedio del tiempo de espera de alguien que quiere ambos platos
- $\bar{W}$: promedio del tiempo de espera de todos los clientes. (Esta es una adición nuestra)

Este sistema se puede modelar como una red de **Jackson abierta**. Donde cada cola es **M/M/1** y hay un backdrop desde cada servidor a la otra cola con probabilidad p_i.

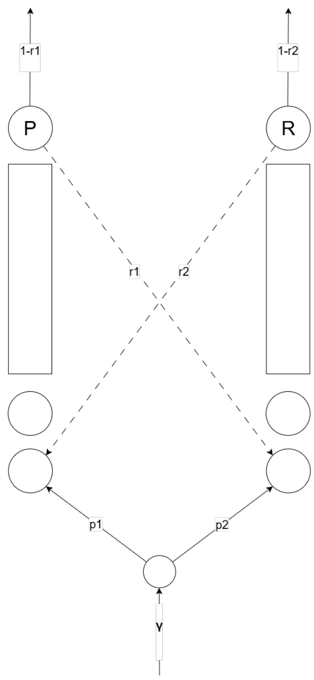


Figura 1 – Modelo red de Jackson abierta

1.3. Variables que describen el problema

- γ: Tasa de llegada de clientes, que es una distribución de Poisson con una media de 20 clientes por hora.
- p_1: Porcentaje de clientes que van a por rollitos de primavera, que es del 60
- p_2: Porcentaje de clientes que van a por pollo frito, que es del 40
- r_1: Porcentaje de clientes que, después de comprar rollitos, van a por pollo, que es del 20
- r_2: Porcentaje de clientes que, después de comprar pollo, van a por rollitos, que es del 10
- μ_1: El tiempo de servicio del cada rollito distribuye exponencial con media 4 minutos.
- μ_2: El tiempo de servicio del cada pollo frito distribuye exponencial con media 5 minutos.

2. DETALLES DE IMPLEMENTACIÓN

Como entorno para implementar las simulaciones utilizaremos **Jupyter Notebook** y como lenguaje de programación **python**. Decidimos utilizar Jupyter por la interactividad permite y la modularidad en la ejecución del código.

Usamos dos modelaciones diferentes para realizar los estudios propuestos. La primera es la que modela el problema original, sin ninguna alteración en las variables y estructuras se definen en la orden, esta es la correspondiente al método **Sim1**. La segunda en la que modelamos prioridad en las personas que piden más de un plato es la correspondiente la método **Sim2**.

Para poder llevar a cabo la simulación, tomamos la decisión de simular las variables aleatorias necesarias a mano (sin usar ninguna librería de terceros), con el objetivo de poner en práctica los conocimientos adquiridos en el curso de probabilidades. Luego solo queda correr el método **ComputeValues** con ambas simulaciones y obtener los resultados

2.1. Estructuras de datos

Como estructuras de datos auxiliares tenemos las clases:

- **Client**: esta clase sirve para almacenar la información necesaria de cada cliente nos permitirá tanto computar datos, como decidir acciones realizar en la simulación.
    - **arrivalP**: tiempo de llegada a la cola del pollo frito, si lo hace.
    - **departureP**: momento de salida a la ventana del pollo frito, si lo hace.
    - **arrivalR**: tiempo de llegada a la cola de los rollitos, si lo hace.
    - **departureR**: momento de salida a la ventana de los rollitos, si lo hace.
    - **bothMeals**: este campo en **Sim1** se utiliza para determinar si el cliente consumió ambas comidas y en **Sim2** se computa a la llegada del cliente para saber durante la ejecución si este cliente quiere ambos platos.
    - **isFirstMeal**: premite conocer si el cliente ya tomó su primer plato, en caso de quiera ambos.
    - **setSecondMeal()**: este método establece **isSecondMeal** a falso, indicando ya tomó su segundo plato.
- **Event**: esta clase sirve para almacenar la información referente a cada evento.
    - **eventName**: aquí se guarda el identificador del evento, los posibles valores se definen en 2.1.1.
    - **time**: momento en ocurre el evento.
    - **clientId**: el identificador del cliente afectado por el evento en cuestión.

2.1.1. Nombres de enventos

- Llegada a la cola del pollo frito.
- Llegada a la ventana del pollo frito.
- Salida de la ventana del pollo frito.
- Llegada a la cola de los rollitos.
- Llegada a la ventana de los rollitos.
- Salida de la ventana de los rollitos.
- Cambio a la cola del pollo frito.
- Cambio a la cola de rollitos.

2.2. Métodos auxiliares

- **ComputeValuesInSim(clientInfo, maxTime)**: Este método recibe un arreglo con los clientes y la duración de la simulación. Devuelve calculadas las variables, cantidad total de clientes, $\bar{W}, \bar{W_P}, \bar{W_R}, \bar{W_{PR}}$ en al simulación correspondiente.

- **ComputeVar(Sim, index, iters, simulationResults, d, time)**: Este método recibe el método que realiza la simulación, el índice en el objeto retorna la simulación de la variable que queremos estimar, la cantidad de iteraciones realizadas hasta el momento, los resultados de la simulaciones anteriores, el valor de d y el tiempo que dura una simulación. Este método implementa el algoritmo propuesto en (ROSS, 2012), para estimar la variable en cuestión y saber cuando se ha simulado suficiente. Devuelve la variable estimada y la cantidad de iteraciones que se han realizado.

- **ComputeValues(Sim, time)**: Este método es quien corre las simulaciones y calcular las variables de interés, utilizando el método **Sim** y **ComputeVar** respectivamente. Corre primero un número fijo de simulaciones, en este caso 100, y las va almacenando en **simulationResults** y ejecuta por cada variable el método **ComputeVar**. Este además de calcular la variable, añade más simulaciones a **simulationResults** para ser reutilizadas en el computo de las demás varaibles.

2.3. Sim1: Problema original

El método **Sim1** recibe como parámetros las variables aleatorias definen el comportamiento de la simulación, con el objetivo de poder cambiar los parámetros de de estas variables sin afectar la implementación de la simulación. Así como la duración de la simulación en minutos. Y nos retorna: la cantidad de clientes atendidos y los valores de las variables $\bar{W}, \bar{W_P}, \bar{W_R}, \bar{W_{PR}}, \bar{L}$.

Primeramente se inicilaizan las variables de estado de la simulación

- **time**: matiene el tiempo actual de la simulación

- **aT**, **dTP**, **dTR**: guardan cuando ocurrirá la próxima llegada y el momento en que termina el servicio a los clientes en las ventanas de pollo frito y rollitos respectivamente.

- **lastClientId**, **clientCount** cuantos clientes han pasado por el sistema, en total, y cuantos hay en el momento, respectivamente.

- **L**, **LCount**: la sumatoria necesaria para computar $\bar{L}$ y cuantas muestras se han tomado.

- **qR**, **qP**: las colas del sistema

- **isFreeP**, **isFreeR**: indican si el servidor está ocupado o no.

- **clientOnP**, **clientOnR**: el id del cliente actualmente en el servidor.

- **eventList**: la lista de eventos.

- **clientInfo**: la lista de clientes.

Luego el flujo es el siguiente: La simulación corre en bucle ejecutando el evento más próximo, uno a la vez, hasta que se alcanza el tiempo total.

Si lo próximo es un arribo, la variable aleatoria **POrR** nos indica a que cola va, se actualizan los estados correspondientes y se añaden el evento correspondiente a la lista y el cliente a la lista de clientes. Si no hay nadie en el servidor correspondiente y hay personas en la cola, la primera pasa a la ventana se añade el evento y se actualizan los estados.

Si lo próximo es que termina, es que alguno de los servidores terminó su pedido. El cliente actual sale. Si las variables **FromPToR** o **FromRToP** lo indican, este cambia a la otra fila si no a tomado ambos platos ya, si no, sale del sistema. Si en la cola hay clientes, el próximo pasa al servidor, se añade el evento y se actualizan los estados.

Al terminar la simulación se computan las variables de interés usando el método **ComputeValuesInSim** y se retornan.

2.4. Sim2: Problema original

Aquí modelamos el hecho de darle prioridad a los clientes que quieren ambos platos. Por simplicidad en la implementación lo que hicimos fue añadir otra cola a cada servidor. Esta segunda cola es solo para clientes que quieren ambos platos. No pasan clientes de la cola normal hasta que la cola de los clientes con ambos platos esté vacía. El objetivo de dicha modelación es comprobar si esto disminuye los tiempos medios de espera.

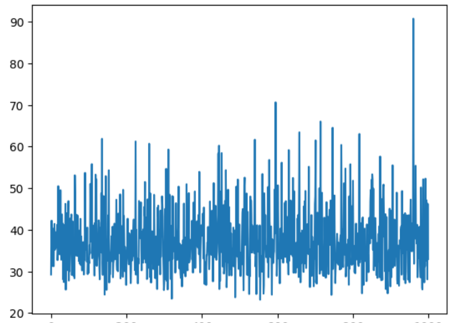
qRBoth, **qPBoth**: estas son las nuevas colas quese añaden al sistema.

No cambian ni los posibles eventos ni el flujo general de la simulación.

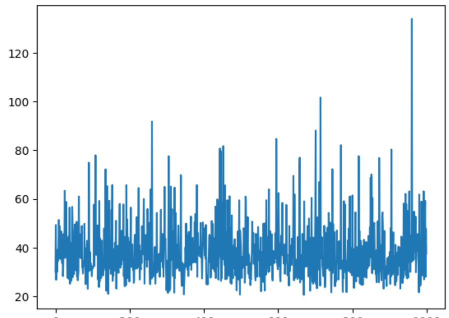
3. RESULTADOS Y EXPERIMENTOS

3.1. Primer experimento con la simulación

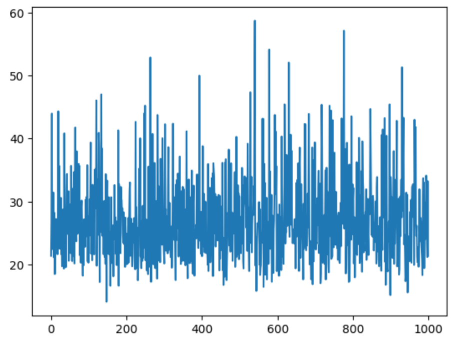
Luego de correr 1000 simulaciones las variables se comportaron de esta manera:



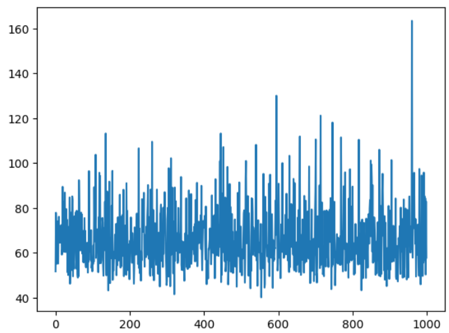
Variable $\bar{W}$



Variable $\bar{W_R}$



Variable $\bar{W_P}$)



Variable $\bar{W_{PR}}$

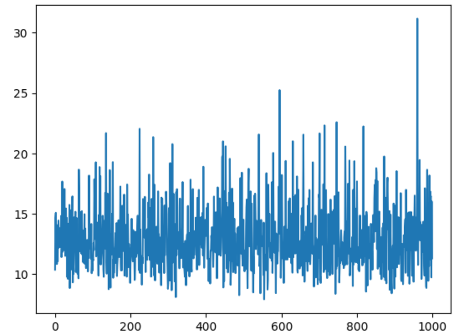


Figura 2 – Variable $\bar{L}$

3.2. Análisis de la condición de parada

Este análisis lo hicimos basándonos en la teoría desarrollada en (ROSS, 2012). Luego como necesitamos computar más de un valor a la vez de la misma simulación, una simulación puede ser costosa computacionalmente y cada variable tiene una condición de parada diferente, aplicamos la técnica explicada en 2.2.

Decidimos seleccionar d(desviación estándar del estimador) de 10 segundos para las medias relacionadas con tiempo y de 1 persona para las medias relacionadas con personas (valga la redundancia). Visto como se comportaban los datos en 3.1 nos pareció que podíamos ser bastante precisos en las estimaciones por eso escogimos d pequeños en cada caso.

3.3. Resultados

Al correr la simulación hasta hasta que salte la condición de parada obtuvimos los siguientes valores estimados para las variables de interés.

Cuadro 1 – Resultados Sim1

Variable	**Estimación**	**Varianza**	**Iteraciones**
$\bar{W}$	36.945513181781756	60.17375128078904	2169
$\bar{W_P}$	37.882439111896105	131.64129950894778	4743
$\bar{W_R}$	26.85446015676669	46.6244740628391	4743
$\bar{W_{PR}}$	65.55166730328183	190.70831584260708	6872
$\bar{L}$	12.826713655811783	7.537819269082322	6872

Con sus respectivas desviaciones.

3.4. Primera Hipótesis

A partir de los datos obtenidos, sobre todo de los valores de $\bar{W}$ y $\bar{W_{PR}}$. Creemos que si se le asignara prioridad de alguna manera a los clientes que quieren ambos platos el promedio de tiempo de espera disminuirá.

Para realizar el experimento es que se decide implementar **Sim2**. Luego de correr **ComputeValues** con esta nueva simulación obtenemos los siguientes valores.

Cuadro 2 – Resultados Sim2

Variable	Estimación	Varianza	Iteraciones
$\bar{W}$	37.360526382215674	61.19358883653936	2206
$\bar{W_P}$	38.17662984016399	134.3668250485684	4842
$\bar{W_R}$	27.082631855193046	49.01997395597875	4842
$\bar{W_{PR}}$	20.610271815867748	0.6375531655726252	4842
$\bar{L}$	12.919552608766438	7.714398362630581	4842

Es válido notar que ambos conjuntos de variables tienen valores muy similares, excepto $\bar{W_{PR}}$ que es bastante intuitivo que disminuya. Luego es necesario realizar un prueba de hipótesis para verificar la diferencia o no de las medias.

3.4.1. Prueba de la hipótesis

Dada la hipótesis definida anteriormente, tomemos los parámetros con subíndice 1 como los referentes a los tomados del problema original y con subíndice 2 los referentes al experimento. Entonces procedemos como sigue.

Se conoce que en las pruebas de hipótesis relacionadas con este campo se utilizan niveles de confianza del 95 %, luego $\alpha = 0.05$.

$$H_0 : \bar{W}_1 - \bar{W}_2 = 0$$
$$H_1 : \bar{W}_1 - \bar{W}_2 > 0$$

Dado que la muestra es bastante grande e independientes por al amanera en que se generan, podemos usar el siguiente estadístico de prueba, distribuye normal.

$$Z = \frac{(\bar{w}_1 - \bar{w}_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Procedemos a calcular (el tamaño de la muestra es la cantidad de iteraciones)

$$Z = \frac{(36{,}945513181781756 - 37{,}360526382215674}{\sqrt{\frac{60{,}17375128078904}{2169} + \frac{61{,}19358883653936}{2206}}} = -1{,}7619148860646704278033819709783$$

Esta es una prueba de cola derecha, por lo que hay un solo valor crítico.

$$Z_\alpha = Z_{0{,}05} = 1,960$$

Luego la región de rechazo es $[1{,}960, \infty)$.

Como -1.7619148860646704278033819709783 $<$ 1,960. No tenemos evidencia suficiente para rechazar la hipótesis nula por lo cual se acepta H_0: $\bar{W}_1$ - $\bar{W}_2 = 0$.

De aquí entendemos que a pesar de que en el modelo original los clientes que quieren ambos platos tienen $\bar{W_{PR}}$ mucho mas alto que $\bar{W}$, la cantidad de estos clientes no es lo suficientemente grande como para afectar la media. De lo que surge la idea de otro Experimento, este seria probar con probabilidades de cambiar de fila mayores pero al intentar el experimento el tiempo que demora la simulación aumenta de manera drástica imposibilitando el experimento, con las tecnologías de las que disponemos.

4. MODELO MATEMÁTICO

4.1. Modelo matemático

Como se menciono en la sección 1.2, nuestro problema se puede modelar como una red de Jackson abierta. Donde tenemos dos colas M/M/1 y ambas tienen un backdrop hacia la otra. Las variables que describen nuestra red son. (SABATER, 2015/2016)

- $\gamma_R = 12$
- $\gamma_P = 8$
- $\mu_R = 15$
- $\mu_P = 12$

$r_{i,j}$	0	P	R
P	$\frac{9}{10}$	0	$\frac{1}{10}$
R	$\frac{8}{10}$	$\frac{2}{10}$	0

Cuadro 3 – valores de $r_{i,j}$

En una red de **Jakson abierta** se conoce como λ_i a la razón de llegada de clientes a la cola i y se calcula de la siguiente manera.

$$\lambda_i = \gamma_i + \sum_{j=1}^{k} \lambda_j r_{ji}$$

Luego:

$$\lambda_R = \gamma_R + \lambda_P r_{PR} = 12 + \frac{1}{10}\lambda_P$$

$$\lambda_P = \gamma_P + \lambda_R r_{RP} = 8 + \frac{2}{10}\lambda_P$$

$$\lambda_R = \frac{640}{49} = 13{,}061224489795918367346938775 51$$
$$\lambda_P = \frac{520}{49} = 10{,}612244897959183673469387755102$$

Entonces ahora podemos calcular las variables de interés de nuestro para nuestro modelo. Procedamos de la siguiente manera. Primero obtengamos ρ_R y ρ_P conocidos como probabilidad de q el sistema este ocupado.

$$\rho_R = \frac{\lambda_R}{\mu_R} = \frac{13{,}061224489795918367346938775 51}{15} = 0{,}870748299319727891156462585034 01$$

$$\rho_P = \frac{\lambda_P}{\mu_P} = \frac{10{,}612244897959183673469387755102}{12} = 0{,}884353741496598639455782312925 17$$

Ahora podemos calcular L_R y L_P que nos permitirán obtener $\bar{L}$.

$$\bar{L_R} = \frac{\rho_R}{1-\rho_R} = \frac{0{,}8707482993197278911564625850 3401}{0{,}1292517006802721088435374149 6599}$$
$$\bar{L_R} = 6{,}736842105263157894736842105 2629$$

$$\bar{L_P} = \frac{\rho_P}{1-\rho_P} = \frac{0{,}884353741496598639455782312 92517}{0{,}115646258503401360544217687 07483}$$
$$\bar{L_P} = 7{,}647058823529411764705882352 9412$$

$\bar{L} = \bar{L_R} + \bar{L_P}$
$\bar{L} = 6{,}7368421052631578947368421052629 + 7{,}6470588235294117647058823529412$
$\bar{L} = 14{,}383900928792569659442724458204$

Y por último los promedios de tiempo de espera $\bar{W}$, $\bar{W_R}$, $\bar{W_P}$, pero no disponemos de teoría para poder calcular $\bar{W_{PR}}$. Esto no quiere decir q no es posible, sino que durante nuestra investigación no la hallamos.

$$\bar{W_R} = \frac{L_R}{\lambda_R} = \frac{6{,}7368421052631578947368421052629}{13{,}061224489795918367346938775 51} = 0{,}51578947368421052631578947368 42$$

$$\bar{W_P} = \frac{L_P}{\lambda_P} = \frac{7{,}6470588235294117647058823529412}{10{,}612244897959183673469387755102} = 0{,}72058823529411764705882352941177$$

$$\bar{W} = \frac{\bar{L}}{\gamma_R + \gamma_P} = 0{,}71919504643962848297213622291 02$$

4.2. Comparativa de resultados

La siguiente tabla relaciona los valores obtenidos en la simulación y en el modelo matemático. Para poder trabajar con las mismas unidades de medida fue preciso multiplicar por 60 los valores de las medias de tiempo.

Variable	Simulación	Modelo
$\bar{L}$	12.826713655811783	14,383900928792569659442724458204
$\bar{W_P}$	37.882439111896105	30.947368421052631578947368421052
$\bar{W_R}$	26.85446015676669	43.235294117647058823529411764706
$\bar{W}$	36.945513181781756	43.151702786377708978328173374612

Cuadro 4 – Comparativa de resultados

Como se puede notar los valores del modelo y la simulación difieren, nos queda propuesto como tarea estudiar el motivo.

REFERENCIAS

ROSS, S. M. *Simulation, Fifth Edition*. [S.l.]: Academic Press, 2012.

SABATER, J. P. G. *Aplicando Teoría de Colas en Dirección de Operaciones*. [S.l.]: Grupo ROGLE, Departamento de Organización de Empresas, Universidad Politécnica de Valencia, 2015/2016.