

Métodos de Modelado y Optimización

Laboratorio 3

Descripción:

La simulación de colas nos permite predecir el comportamiento que sistemas computacionales o reales de colas van a tener bajo circunstancias dadas. Como parte de este laboratorio diseñaremos un simulador de un sistema de colas paramétrico que nos permitirá estimar métricas relacionadas al sistema de colas simulado.

Implemente las siguientes estructuras solicitadas en lenguaje Python.

1. (10%) Método `random_exp(lambd)` que retorna un número aleatorio con distribución exponencial.
 - a. Para ello podemos aplicar [inverse transform sampling](#) que nos permite convertir un número aleatorio con distribución uniforme $[0,1[$ en un número de otra distribución probabilística.
 - b. En el caso de la distribución exponencial la ecuación sería:
$$-\ln(1 - \text{rand}[0,1[) / \lambda$$
2. (10%) Implemente una clase `Client` que simule ser un cliente del sistema. Al llegar a la cola el cliente contará el tiempo que estuvo esperando en la cola y posteriormente el tiempo tomado en el proceso de servicio. Estas métricas serán usadas para calcular el tiempo de espera promedio (en cola y en servicio) de los clientes.
3. (10%) Implemente una clase `Server` que simule ser un servidor del sistema. Al iniciar la cola se cuenta con uno o más servidores en el sistema los cuáles no estarán atendiendo clientes hasta que haya alguno en la cola. Cuando terminan con un cliente, inmediatamente pasan al siguiente (asumiendo que haya alguno en cola). Además cada servidor contará el tiempo que estuvo desocupado. Esta métricas se utilizará para calcular el tiempo promedio de desocupación para los servidores.
4. (20%) Implemente una clase `Queue` que recibe una longitud máxima de fila `lmax`, una cantidad de servidores `s`, una función `lambd` que calcula la tasa de llegada de clientes dado `n` (`n` siendo la cantidad de clientes en el sistema) y una función `mu` que calcula la tasa de servicio de clientes dado `n`.
 - a. (40%) La clase `Queue` debe contener un método `simulation(time_limit)` que ejecuta una simulación para la cola dada, asumiendo que el tiempo empieza en 0 (hora en la que se “abre” la cola y se empieza a esperar la primer llegada) y `time_limit` (hora en la que se “cierra” la cola, pero sin embargo se atienden los clientes restantes). Al terminar la simulación debe imprimir: total de clientes servidos durante el intervalo, total de clientes perdidos (llegaron y la cola ya estaba llena), tiempo de espera promedio en la

cola, tiempo de servicio promedio para los clientes y tiempo promedio ocioso entre los servidores.

- b. Sugerencias: El tiempo no es necesario manejarlo con un contador de tiempo real (sino las simulaciones podrían tardar mucho). En lugar de eso puede usar una cola de eventos: cada vez que llega un cliente, se genera un nuevo evento de la siguiente llegada de cliente, con tiempo de llegada $\text{tiempo_actual} + \text{random_exp}(\lambda(n))$ a la cola de eventos. Cada vez que un cliente es atendido por un servidor, se agrega un evento de fin de servicio de cliente, con tiempo de servicio $\text{tiempo_actual} + \text{random_exp}(\mu(n))$. De esta manera podemos “avanzar” el tiempo de manera iterativa, tomando siempre el evento siguiente más próximo y eliminándolo de la cola. Cuando ocurre un evento de llegada de cliente, se agrega un nuevo cliente a la cola, y se genera el nuevo tiempo de llegada. Cuando ocurre un evento de finalización de servicio de cliente, el servidor correspondiente termina de atender al cliente y queda libre para atender al siguiente, si atiende un nuevo cliente, se vuelve agregar un nuevo evento de finalización a la cola de eventos.
5. (10%) Documento escrito. Utilice su sistema para analizar el caso de una empresa que posee una cola de tamaño máximo 15, un λ dado por la función $\lambda(n) = 64 - n^{1.5}$ y un μ dado por la función $\mu(n) = 5 + 3n$ con 1 servidor. Calcule el comportamiento que se esperaría después de 1000 horas de tener la cola en funcionamiento. Repita el experimento 3 veces para tomar múltiples mediciones y documente los datos obtenidos. ¿Cuál fue el promedio de clientes servidos? ¿Cuál fue el promedio de clientes que se perdieron por no poder ingresar a la cola? ¿Cuál fue el tiempo de espera en la cola promedio para los clientes? ¿Cuál fue el promedio de tiempo ocioso para el servidor? Luego repita el experimento con 2 y 3 servidores. ¿Cuál de los modelos logra un buen balance entre el tiempo de espera y el tiempo ocioso? ¿Qué recomendaría usted y por qué?

Nota: Se permite utilizar la biblioteca de soporte math, numpy, pero no se permite utilizar bibliotecas que realicen el trabajo por usted. Cualquier otra biblioteca que se desee usar debe ser consultada primero con el profesor.