

Proyecto de Simulación
Aldo Verdesia Delgado C-412

Orden del problema:

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.

Ideas de Solución:

Como primer enfoque surgió la idea de ir generando entradas mientras el tiempo estuviera disponible, y sacar los pedidos al llegar al tiempo de terminar alguno. Cree una clase Sistema la cual contenía, los estados de las elaboraciones, se tiene en esa estructura que pedido esta haciendo cada cocinero y sus respectivos tiempo de finalización, así al terminar un cocinero escoja automáticamente el próximo pedido en lista, garantizando que todos sean atendidos. Seguí el esquema de cuando generaba una nueva entrada y aun estaba sin finalizar alguno de los pedidos que estaban elaboración los sumaba al sistema, en el momento que el tiempo de la próxima entrada fuera mayor que el de terminar de algún cocinero entonces atendía esa salida y corría el tiempo real hasta la suma del tiempo anterior con la salida. En el momento que ya no tenia mas arribos entonces me aseguro de vaciar la cola y darle salida a todos los cliente.

Inicialice el problema de la siguiente manera. Cree en cero variables como tiempo , tiempo de terminar de cada cocinero y la que contabiliza cada entrada.

Luego tenia los siguientes eventos :

Arribaba un cliente : pues lo sumaba a la lista y luego le asignaba si un cocinero de estar disponible sino a la cola

Salida del primer cocinero N : el tiempo de partida de ese cliente seria el tiempo transcurrido mas el tiempo en que termino su pedido,luego busca el primer cocinero el próximo pedido de la lista de pendientes

Cierre : debo vaciar el sistema utilizando , debido a que el tiempo de llegada de cualquier cliente ya supera el tiempo permitido , entonces empezaría a generar la salidas de cada uno de los n cocinero , terminaría en el momento que la cantidad de pedidos pendientes sea 0

Análisis :

Al correr mi algoritmo,con dos cocineros cuatro veces consecutivas para ver los distintos tipos de valores me dio el resultado siguiente , para valores de lambda 0,1 en horario normales y 0.5 en pico

El % de los que esperan mas de 5 min es 59.333333333333336

El % de los que esperan mas de 5 min es 76.92307692307693

El % de los que esperan mas de 5 min es 69.28104575163398

El % de los que esperan mas de 5 min es 54.794520547945204

Aqui puede apreciarse que ampliamente con la utilización de 2 cocinero hace muy probable que mas del 50% de los clientes esperen mas de 5 min , y que se realicen varias colas en los distintos momentos del dia

Al probar para valores de lambda 0,3 en horario normales y 0.7 en pico la salida es la siguiente :

El % de los que esperan mas de 5 min es 88.02816901408451

El % de los que esperan mas de 5 min es 91.63636363636364

El % de los que esperan mas de 5 min es 87.59398496240601

El % de los que esperan mas de 5 min es 94.69964664310953

Por lo visto mientras mas frecuentes sea el arribo de cliente menos dan a vasto los cocineros haciendo que la cola y la espera sea mayor :

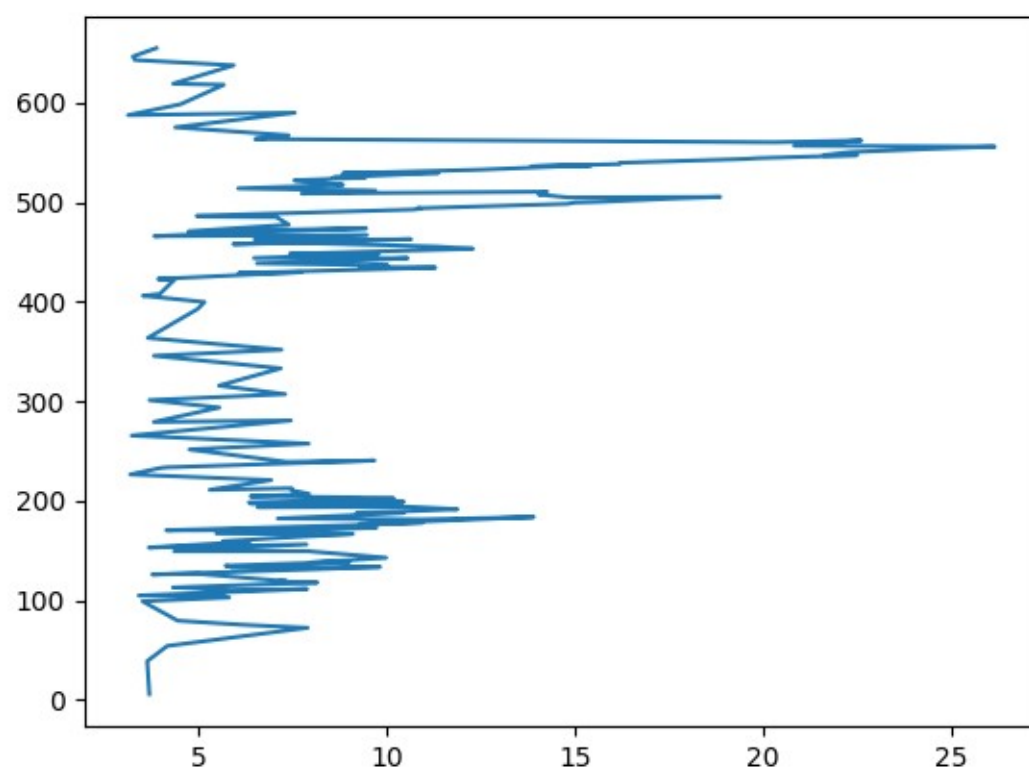
Al hacer el proceso incorporando un nuevo cocinero en la hora pico , la variación en la salida para los mismos valores de lambda son bastantes considerables por ejemplo para valores de lambda 0,1 en horario normales y 0.5 en pico

El % de los que esperan mas de 5 min es 8.904109589041095

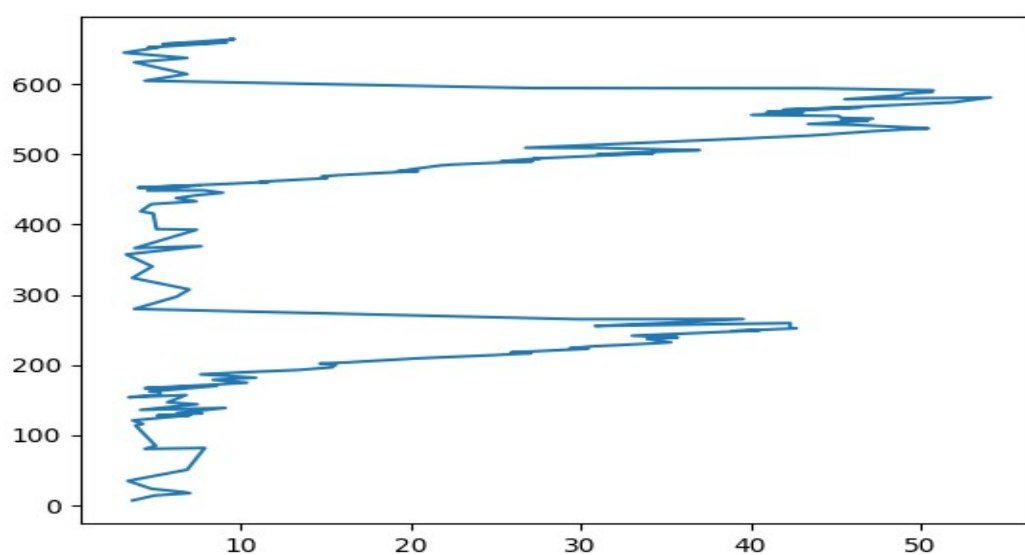
El % de los que esperan mas de 5 min es 24.11764705882353

El % de los que esperan mas de 5 min es 2.4

El % de los que esperan mas de 5 min es 4.964539007092199



grafica 1



En la grafica 1 se puede apreciar como es el comportamiento desde el tiempo que llegan hasta que se van , cuando trabaja otro cocinero en los tiempos picos. En la grafica 2 se muestra la mismo pero para el otro caso.

Conclusiones:

En el restaurante por lo visto , con las diferentes simulaciones , es bastante obvio que la presencia de otro cocinero en el horario pico mejora mucho el tiempo de espera de los clientes , y a menor frecuencia de la llegada de los clientes mejor , mejora también la eficiencia de los pedidos

Especificaciones del código :

El programa principal corre llamando al método Kojo ubicado en cocina.py. Este método tiene un parámetro bool el cual si es true corre con solo 2 cocinero y en caso contrario con 3