

Proyecto Eventos Discretos

Overloaded Harbor

ALEJANDRO CAMPOS. C-411

Facultad de Matemática y Computación
Universidad de la Habana
2021

I. INTRODUCCIÓN

La simulación por eventos discretos es una técnica de modelado dinámico de sistemas. Esta se caracteriza por un control en la variable del tiempo que permite avanzar a este a intervalos variables, en función de la planificación de ocurrencia de tales eventos a un tiempo futuro. Un requisito para aplicar esta técnica es que las variables que definen el sistema no cambien su comportamiento durante el intervalo simulado.

El problema de eventos discretos seleccionado para su resolución es Overloaded Harbor, cuya orden se detalla a continuación:

En un puerto de supertanqueros que cuenta con 3 muelles y un remolcador para la descarga de estos barcos de manera simultánea se desea conocer el tiempo promedio de espera de los barcos para ser cargados en el puerto.

El puerto cuenta con un bote remolcador disponible para asistir a los tanqueros. Los tanqueros de cualquier tamaño necesitan de un remolcador para aproximarse al muelle desde el puerto y para dejar el muelle de vuelta al puerto.

El tiempo de intervalo de arribo de cada barco distribuye mediante una función exponencial con $\lambda = 8$ horas. Existen tres tamaños distintos de tanqueros: pequeño, mediano y grande, la probabilidad correspondiente al tamaño de cada tanquero se describe en la tabla siguiente. El tiempo de carga de cada tanquero depende de su tamaño y los parámetros de distribución normal que lo representa también se describen en la tabla siguiente.

Tamaño	Probabilidad de Arribo	Tiempo de Carga
Pequeño	0.25	$\mu = 9, \sigma^2 = 1$
Mediano	0.25	$\mu = 12, \sigma^2 = 2$
Grande	0.5	$\mu = 18, \sigma^2 = 3$

De manera general, cuando un tanquero llega al puerto, espera en una cola (virtual) hasta que exista un muelle vacío y que un remolcador esté disponible para atenderle. Cuando el remolcador

está disponible lo asiste para que pueda comenzar su carga, este proceso demora un tiempo que distribuye exponencial con $\lambda = 2$ horas. El proceso de carga comienza inmediatamente después de que el barco llega al muelle. Una vez terminado este proceso es necesaria la asistencia del remolcador (esperando hasta que esté disponible) para llevarlo de vuelta al puerto, el tiempo de esta operación distribuye de manera exponencial con $\lambda = 1$ hora. El traslado entre el puerto y un muelle por el remolcador sin tanquero distribuye exponencial con $\lambda = 15$ minutos.

Cuando el remolcador termina la operación de aproximar un tanquero al muelle, entonces lleva al puerto al primer barco que esperaba por salir, en caso de que no exista barco por salir y algún muelle esté vacío, entonces el remolcador se dirige hacia el puerto para llevar al primer barco en espera hacia el muelle vacío; en caso de que no espere ningún barco, entonces el remolcador esperará por algún barco en un muelle para llevarlo al puerto. Cuando el remolcador termina la operación de llevar algún barco al puerto, este inmediatamente lleva al primer barco esperando hacia el muelle vacío. En caso de que no haya barcos en los muelles, ni barcos en espera para ir al muelle, entonces el remolcador se queda en el puerto esperando por algún barco para llevar a un muelle.

Simule completamente el funcionamiento del puerto. Determine el tiempo promedio de espera en los muelles.

El objetivo de este informe es explicar el método de resolución utilizado para resolver el problema anterior. Además explicaremos los detalles de implementación y ejecución de la aplicación, no sin antes abordar el modelo de eventos discretos utilizado y cómo se adaptó a este problema en particular. Por último, haremos un análisis de los resultados obtenidos a partir de la ejecución de las simulaciones del problema.

II. IDEAS PRINCIPALES PARA LA SOLUCIÓN DEL PROBLEMA

Para tratar este problema se siguieron las indicaciones de conferencia y del libro "Temas de simulación", capítulo 3. Estos indican, en forma de resumen, que los elementos básicos de una simulación basada en eventos discretos son las variables y los eventos. Luego cuando ocurra un evento los valores de las variables se actualizan y se genera el tiempo de ocurrencia del próximo evento. Posteriormente, se selecciona para ejecutar el evento cuyo tiempo sea menor.

En el caso de este problema, cuando un tanquero arriba al puerto se genera el tiempo del próximo arribo y se inserta en la lista de espera para ser movido a un muelle. Esta lista está ordenada por el tiempo en que el tanquero ejecutará su próximo evento, el siguiente evento a ejecutar será aquel cuyo tiempo sea menor y se pueda ejecutar según las condiciones del problema, ya que cada tanquero para moverse debe esperar a que el remolcador esté disponible, y que haya algún muelle vacío, en caso de que el tanquero esté en el puerto. La forma de escoger el siguiente evento a procesar evita que un barco espere más del tiempo que debería y evita, además, que un barco se quede esperando eternamente. Una vez que se pueda trasladar el barco hacia un muelle, se genera el tiempo del próximo evento, que sería la llegada del tanquero al muelle y el comienzo de la carga. De esta forma cada evento determina el tiempo del siguiente hasta que el tanquero abandona la simulación, guardando el tiempo de estancia desde su llegada hasta su partida antes de salir. Se tienen entonces los siguientes eventos a simular:

1. Llegada de un tanquero al puerto.

2. Traslado de un tanquero a un muelle si hay alguno disponible y el remolcador no está ocupado.
3. Llegada de un tanquero al muelle e inicio de la carga.
4. Momento en que un tanquero terminó de cargar y está listo para regresar al puerto cuando el remolcador esté disponible.
5. Traslado de un tanquero del muelle al puerto.
6. Llegada del tanquero al puerto y partida de este.

Podemos mencionar, además, como evento adicional pero no correspondiente a los tanqueros, el movimiento del remolcador vacío hacia o desde los muelles en caso de que no hayan barcos esperando en el lugar donde se encuentra.

Por otro lado, debemos tener en cuenta a lo largo de la simulación: el tiempo general de esta, el tiempo de arribo y de partida de cada tanquero y el tiempo del próximo evento del mismo, constituyendo estas nuestras variables de tiempo. Las variables de estado, que describen el estado del puerto en el tiempo t , serían la localización del remolcador (en el puerto, en el muelle o en movimiento) y la cantidad de muelles libres. Por último, la cantidad de arribos y la lista de los tiempos que los tanqueros han estado en el puerto son nuestras variables contadoras.

Las variables anteriores constituyen la base de la simulación implementada, su uso concreto en el código se verá con mejor detalle en la sección IV del presente informe.

III. MODELO DE SIMULACIÓN DE EVENTOS DISCRETOS

Para simular el problema se usó como base el modelo de eventos discretos de n servidores en paralelo, o sea, cuando llega un cliente, si hay un servidor vacío se atiende a este, si todos están llenos, se pone en la cola a esperar que uno se vacíe. En este problema los clientes serían los tanqueros y los servidores los muelles.

En el caso de este problema los clientes deben ser desplazados, por orden de llegada, a los servidores. Esta operación debe realizarse de uno en uno, puesto que solo se cuenta con un remolcador. Los clientes son atendidos por los servidores y posteriormente deben ser trasladados, también de uno en uno, hacia la salida. Solo puede existir a lo sumo un cliente desplazándose en el mismo intervalo de tiempo. El orden en que son desplazados a la salida depende del tiempo que demoren en llegar al servidor y lo que tarde el servidor en atenderlos (cargando el tanquero).

La estancia de cada cliente en el sistema se calcula como la diferencia entre el tiempo de salida y el de llegada. Una vez concluida la simulación, la media del tiempo de todos los barcos sería la suma de sus tiempos dividido entre la cantidad de clientes que atendió el sistema.

IV. DETALLES DE IMPLEMENTACIÓN

La implementación de la simulación del problema en cuestión se basa en las ideas anteriores. El código está suficientemente comentado como para entender la funcionalidad de cada uno de los

métodos y variables principales. No obstante explicamos en esta sección, a muy groso modo, algunos detalles de implementación.

Tenemos dos clases principales: *Tanker* y *Harbor*. En la primera están las propiedades correspondientes a los tanqueros, y toda la lógica que corresponde a estos. Cada instancia de esta clase contiene el número de identificación de cada tanquero, el tiempo de llegada y de partida; además del tiempo *current_time* en que se ejecutará el evento *Event* asociado a cada barco, los cuales se modifican cada vez que se ejecuta un evento, pues en la ejecución del evento *i* se conoce el siguiente evento *i + 1* a ejecutar y el tiempo de este.

La clase *Harbor* es la clase principal de la simulación, en esta se encuentran implementados cada uno de los eventos antes descritos. Para crear una instancia de esta clase debemos pasarle como parámetro el tiempo límite de la simulación, este es el tiempo a partir del cual ya no entrará ningún tanquero al puerto, pero la simulación continuará hasta atender todos los tanqueros que quedan pendientes en el sistema. El número de muelles puede pasarse también como parámetro a esta clase, de no pasarlo toma por defecto 3 muelles. En esta clase se crea la lista *pending_tankers* que constituye la cola virtual de los tanqueros, la cual se mantiene ordenada durante toda la ejecución de la simulación, atendiendo al tiempo del próximo evento a ejecutar por los tanqueros. Mediante el método *run_simulation* se desarrolla la simulación llamando al próximo evento más cercano en el tiempo que pueda ejecutarse según la disponibilidad del remolcador y de los muelles.

El archivo *utils/utils.py* contiene los métodos necesarios para generar los tiempos de cada evento. En este se encuentran los métodos de generación de variables aleatorias con distribución exponencial con parámetro λ y distribución normal con parámetros μ y σ^2 .

Finalmente, en el archivo *main.py* se crea la instancia de la clase *Harbor* según los parámetros que ingrese el usuario, y se ejecuta la simulación. En la siguiente sección se describe mejor este proceso.

V. EJECUCIÓN DE LA APLICACIÓN

Para ejecutar la aplicación es necesario que el usuario tenga instalado python versión 3.0 o superior. Luego en la ubicación del proyecto debe abrir una consola y escribir `python main.py` y seguir los pasos que se indican.

La aplicación pedirá al usuario que ingrese el tiempo en horas a partir del cual ya no entrará ningún barco en el puerto, no obstante, como ya se ha dicho, la simulación continuará hasta atender a todos los barcos que aún están dentro del sistema. Por lo que, potencialmente, la simulación durará mucho más que lo que indica este tiempo.

Después de ingresar este número se le pedirá al usuario que ingrese el número de muelles con que desea que la simulación se ejecute, de no entrar ningún número se asume por defecto 3 muelles. Por último, se pide el número de simulaciones que se realizarán con los parámetros antes solicitados, si no entra ningún número se realizará solo una simulación. En caso de que ingrese algún dato inválido, la aplicación lo notificará y el usuario deberá ingresar los parámetros nuevamente.

Una vez concluida la o las simulaciones, se muestran el tiempo total, el número de barcos atendidos en ese tiempo y la media de la estancia de los barcos en el puerto. El usuario podrá repetir el proceso presionando la tecla enter o salir escribiendo el comando `exit` en consola.

VI. CONSIDERACIONES OBTENIDAS A PARTIR DE LA EJECUCIÓN DE LAS SIMULACIONES DEL PROBLEMA

Si el usuario realiza 10 o más simulaciones con los datos constantes, se muestran dos gráficas que representan el comportamiento del tiempo de simulación y de la media de la estancia de los tanqueros, según los barcos que se atendieron.

A partir de realizar 30 simulaciones permitiendo la entrada de tanqueros hasta 5 horas y con 3 muelles se obtuvieron los siguientes resultados:

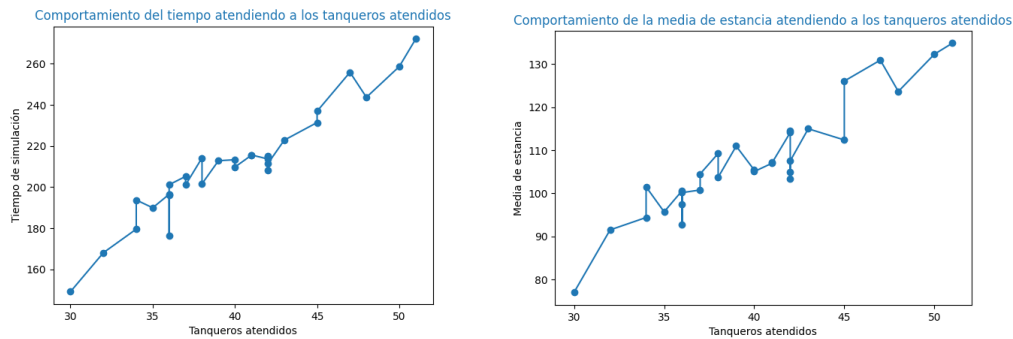


Figure 1

Las gráficas indican que los datos tienen una correlación lineal positiva, lo cual quiere decir que mientras el sistema permita la entrada de más tanqueros, hay una tendencia a que más tiempo se demore en terminar de atenderlos a todos y la media de estancia en el puerto sea mayor.

Por otro lado en 5 horas que se permitió la entrada de tanqueros al puerto se atendieron entre 30 y 55 barcos aproximadamente, esta diferencia viene dada por los diferentes tamaños que pueden tener los tanqueros, que influye en el tiempo de carga, además del tiempo de traslado hacia y desde los muelles, que puede demorarse en dependencia del tiempo generado por la distribución correspondiente. Lógicamente, si aumentamos las horas de entrada al puerto, entrarán más tanqueros y el sistema se demorará más en atenderlos a todos; y la media de espera de los barcos será mayor. Esto se muestra en las gráficas de la figura 2, donde se permitió la entrada de tanqueros hasta 10 horas.

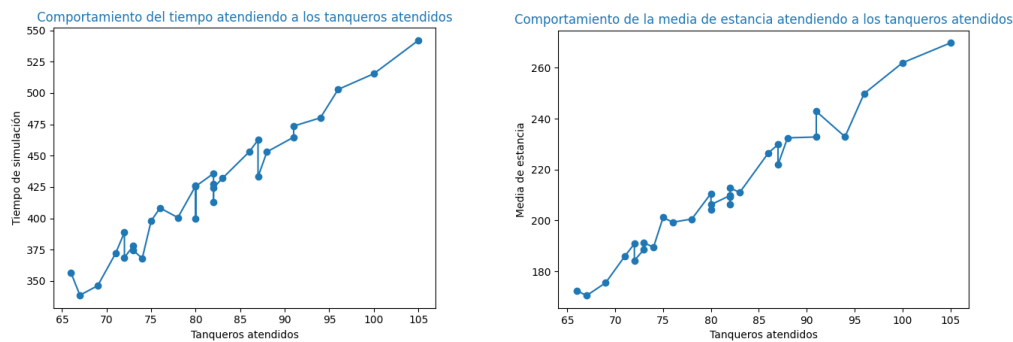


Figure 2

Podemos notar que entraron entre 65 y 105 barcos y que el tiempo de simulación y la media de estancia son mayores con respecto a las gráficas de la figura 1.

Analicemos qué ocurre si aumentamos el número de muelles que pueden atender tanqueros (el número de servidores aumenta). Las gráficas siguientes corresponden a 30 simulaciones permitiendo la entrada de barcos hasta 10 horas en un puerto de 6 muelles.

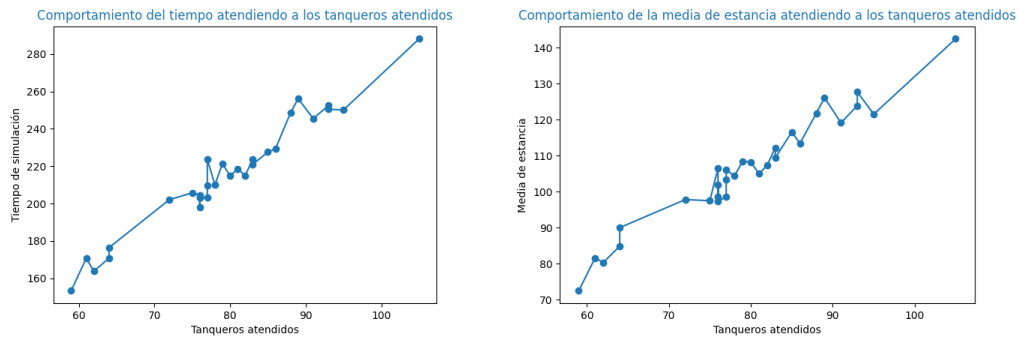


Figure 3

Comparando estos resultados con los de la figura 2, podemos llegar a la conclusión de que, si aumenta el número de muelles, tanto el tiempo de simulación como la media de estancia disminuyen considerablemente. Ya que manteniendo los datos constantes y duplicando el número de muelles notamos que el tiempo de simulación pasa de entre las 350 y 550 horas a estar entre las 160 y 280 horas, tal como indican las gráficas. De igual manera se nota una disminución de la estancia de las naves en el puerto de unas 120 horas aproximadamente como promedio.

VII. GITHUB

En este [enlace](#) podemos encontrar el repositorio del proyecto en Github.