

Simulación por Eventos Discretos, Ejercicios

Resumen— En este reporte se muestra cómo fueron simulados dos problemas planteados en el semestre B-2018 en la Universidad de Los Andes en la materia de Modelado y Simulación 1. Para ambos ejercicios, fue utilizada la biblioteca de R orientada a simulaciones por eventos discretos **Simmer** y para uno de ellos el entorno de simulación **Anylogic**. Para el primer ejercicio, se obtuvieron los resultados expuestos en la tabla #1 de la misma forma, para el segundo ejercicio se determinó que la cantidad de ambulancias necesarias para cumplir el servicio sin la existencia de colas dependerá del valor estipulado en la semilla de simulación, pero normalmente son necesarias en promedio de 6 a 8 ambulancias.

I. INTRODUCCIÓN

La **simulación por eventos discretos** es una técnica informática de modelado dinámico de sistemas. Frente a su homóloga, la simulación de tiempo continuo, esta se caracteriza por un control en la variable del tiempo que permite avanzar a éste 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. En este caso se estudiará la **teoría de colas**, el estudio matemático de las colas o líneas de espera dentro de un sistema. Esta teoría estudia factores como el tiempo de espera medio en las colas o la capacidad de trabajo del sistema sin que llegue a colapsar. Para los análisis y simulaciones se utilizará **R** el cual es un entorno y lenguaje de programación con un enfoque al análisis estadístico y también **Anylogic**, un software de simulación avanzado.

II. RESULTADOS Y ANÁLISIS

Ejercicio 1:

Enunciado:

El mismo del examen teórico, a saber: Los barcos llegan a un puerto con tiempos entre llegada exponenciales con media de 1.25 días. El puerto tiene un muelle con dos atracaderos y dos grúas para descargar los barcos. Los barcos que llegan cuando ambos atracaderos están ocupados se unen a una cola FIFO. El tiempo para que una grúa descargue un barco se distribuye como una uniforme entre 0.5 y 1.5 días. Si solo un barco está en el atracadero, ambas grúas descargan el barco

Este trabajo no fue financiado de ninguna forma.

Morón, J. es Asistente Docente en el Departamento de Computación de la Universidad de Los Andes, Mérida, 5101 VEN (e-mail: juandiegop17@gmail.com).

Rojas, J. es estudiante regular de la Universidad de Los Andes, Mérida, 5101 VEN (e-mail: jesusdanielru1@gmail.com).

Gonzalez, D. es estudiante regular de la Universidad de Los Andes, Mérida, 5101, VEN (e-mail: darwingb18@gmail.com)

y el tiempo se reduce en la mitad. Cuando dos barcos están en el

atracadero, cada grúa se encarga de un barco diferente. Si ambas grúas se encuentran descargando un barco y llega un segundo barco al atracadero una de las grúas comienza a servir a este barco y el tiempo que resta al primero barco se duplica. Corra la simulación por 90 días. Se desea calcular el tiempo promedio de espera de un barco para ser atendido y el tiempo máximo así como también el tiempo mínimo, promedio y máximo de descarga.

Modelo:

El núcleo fundamental del modelo realizado en el lenguaje estadístico R y con la librería de simulación **Simmer**, expuesto en el archivo **pier.R**, consta con una trayectoria principal llamada **boat** y un ambiente llamado **pier**. La trayectoria principal debido a las características del modelo se realizó con tres sub trayectorias mediante el uso de la función **branch** ya que se categorizaron tres tipos de llegadas para los botes. Las tres sub trayectorias siguen el siguiente esquema cambiando solo algunas peculiaridades.

Se asignan tres atributos en la trayectoria, necesarios para la ejecución de las mismas. Tiempo de inicio **start_time**, multiplicador de acuerdo al tipo de llegada **multiplier** y el tiempo de servicio **activity_time**. Luego se realiza un **release** del recurso grúa **crane** donde se ejecuta un **timeout** cuya duración va de acuerdo a una distribución uniforme entre 0.5 y 1.5, por último, se imprime un mensaje que indica que finalizó una sub trayectoria.

En la primera sub trayectoria, el bote llega al puerto y los dos embarcaderos se encuentran desocupados, de acuerdo con el enunciado este caso debe procesarse de manera que mientras no llegue ningún otro barco al sistema, ambas grúas deben atender a este barco, esta sub trayectoria tiene la peculiaridad de que posee un manejador de señal mediante el uso de la función **trap** el cual se dispara en el momento que llegue otro barco al sistema y ambas grúas estén atendiendo a un solo barco, al momento de recibir esta señal el manejador procede a realizar un segundo **timeout** para el primer barco con el fin de simular la condición que dicta

“Si ambas grúas se encuentran descargando un barco y llega un segundo barco al atracadero una de las grúas comienza a servir a este barco y el tiempo que resta al primero barco se duplica”.

La segunda sub trayectoria, simula el segundo caso, cuando llega un bote al sistema y ambas grúas se están ocupando de un único bote, en este caso se envía una señal mediante el uso de la función **send** la cual será recibida como se explicó

anteriormente por la primer sub trayectoria. Se realizo este envío de señal con el fin de determinar el tiempo exacto del segundo **timeout** del primer barco.

La tercera y última sub trayectoria sigue el camino estándar de las otras dos sin ninguna peculiaridad adicional.

El ambiente de simulación **pier** posee dos recursos de tipo **crane** y genera diez barcos de acuerdo a una distribución exponencial con media 1.25. Esta simulación se realizó quince veces con semillas distintas todas para un tiempo de noventa días.

Finalmente, los resultados obtenidos se presentan en una tabla, la cual presentamos a continuación:

tabla #1

	Número de Barcos	Tiempo de espera promedio	Tiempo máximo de espera	Tiempo mínimo de espera	Tiempo de descarga promedio	Tiempo máximo de descarga	Tiempo mínimo de descarga
	10	0.33	2.08	0	3.57	26.46	0.67
	10	0	0	0	0.49	0.67	0.31
	10	0.01	0.11	0	0.6	1.01	0.25
	10	0.21	0.1	0	1.85	9.61	0.52
	10	0.06	0.54	0	3.91	31.45	0.53
	10	0.08	0.75	0	2.61	20.52	0.29
	10	0.02	0.22	0	3.45	28.4	0.37
	10	0.32	1.12	0	2.68	18.39	0.52
	10	0.04	0.24	0	3.01	23.19	0.36
	10	0.06	0.3	0	2.36	16.87	0.42
	10	0.79	2.63	0	2.58	15.39	0.6
	10	0.06	0.64	0	4.6	36.66	0.38
	10	0.16	1.08	0	3.2	25.14	0.31
	10	0.06	0.64	0	3.16	25.75	0.38
	10	0.22	1.04	0	2.07	13.34	0.45
	10	0	0	0	3.92	33.18	0.33
Rango	[10, 10]	[0, 0.79]	[0, 2.63]	[0, 0]	[0.49, 4.60]	[0.67, 36.66]	[0.25, 0.67]
Desviación Estándar	0	0.2	0.73	0	1.12	10.59	0.12

Con los resultados obtenidos en la **tabla #1**, vemos que el modelo se ejecutó de forma correcta cumpliendo con los parámetros expuestos en el enunciado del ejercicio. Con el fin de comprender la simulación que se está realizando se incluye además un archivo **pier_one_run.R** el cual funge como referencia para entender lo que sucede en una sola corrida del sistema.

Ejercicio 2:

Enunciado:

En una larga zona metropolitana, un centro de atención de emergencias recibe solicitudes y despacha ambulancias a una tasa de 1 cada 15 ± 10 minutos. Aproximadamente, 15% de las solicitudes son falsas alarmas que toman 12 ± 2 minutos en ser procesadas. Las restantes solicitudes pueden ser de dos tipos: Los casos graves (15% de las llamadas no falsas) que toman

aproximadamente 25 ± 5 minutos y las llamadas normales (el resto) que tardan 20 ± 10 minutos en procesarse. Asuma que hay un número muy grande de ambulancias y que todas pueden estar disponibles cuando sea necesario. Simule el sistema para 500 llamadas.

a) Estime la cantidad de ambulancias requeridas para cubrir 100% el servicio requerido.

b) Suponga que hay una sola ambulancia disponible. Cualquier llamada que ocurra mientras una ambulancia esta en servicio debe esperar. Puede una sola ambulancia proporcionar el servicio requerido?

c) Simule el sistema con x ambulancias, donde $x = 1, 2, 3$ o 4. Compare las alternativas en terminos de cuanto tiempo debe esperar una solicitud para ser atendida, el porcentaje de solicitudes que deben esperar y el porcentaje de tiempo que las ambulancias están ocupadas.

Modelo:

La simulación para este ejercicio se realizó tanto en el lenguaje estadístico **R**, con librería **simmer** como en **Anylogic**, En el caso de **simmer** se construyó el modelo de la siguiente manera:

Se crearon tres trayectorias **call_1** (falsas alarmas), **call_2** (casos graves) y **call_3** (llamadas normales), una para cada tipo de llamada cada uno da ellas cuanta con un atributo para el tiempo de inicio **start_time** el tiempo en servicio **activity_time** y se realiza un **release**, del servidor **ambulance**, donde se realiza un **timeout** descrito por una distribución uniforme cuyos parámetros difieren de acuerdo al enunciado para cada tipo de llamada. Además, se instancio el ambiente de simulación llamado **hospital**, el cual tiene como recurso ambulancias, con el nombre **ambulance**. El ambiente de simulación genera de acuerdo con los porcentajes en el enunciado, 75 llamadas del tipo **call_1**, 64 llamadas del tipo **call_2** y 361 del tipo **call_3**, todas de acuerdo con una distribución uniforme entre 5 y 25.

En el caso de **anylogic** el modelo se construyó en la versión **Professional 8**, para ello se inició con un **Source** que proporciona la salida en minutos del distinto grupo de llamadas (falsas, graves y normales) con una distribución **uniforme(5,25)** minutos seguidamente se anida a ello un **Select** para la selección de los tipos de llamadas con la siguiente probabilidad: 0,15 para las llamadas falsas, 0,12675 para las graves y 0,7225 para las normales, la selección se reparte según su tipo de llamada a un **Service** (ambulancia) que se encarga de hacer un tiempo de servicio en minutos a cada tipo de llamada, para la falsa llamada una distribución **uniforme(10,14)** minutos, para las graves **uniforme(20,30)** minutos y para las normales **uniforme(10,30)** minutos, los **Services** (ambulancias) prestan su servicio a un cliente (llamada) a la vez, por lo cual se genera una cola. Los **Service** prestan su servicio de acuerdo a la disponibilidad de ambulancias proporcionada por un **ResourcePool** (Cantidad de Ambulancias), los **Services** (ambulancias) prestarán su servicio si y solo si hay **ResourcePools** disponibles en caso contrario se generará un tiempo de espera para la llamada hasta que el **ResourcePool** tenga unidades disponibles, seguidamente están los sinks los cuales son utilizados para la salida de la llamada luego de haber pasado por todo el sistema.

Se contruyeron **4** modelos, el primero para una ambulancia, otro para dos, tres y finalmente un último para cuatro ambulancias. Se puede observar los cambios generados al alterar las unidades disponibles, gráfica y analíticamente.

La simulación se ejecutó de manera distinta para cada inciso obteniendo los mismos resultados tanto en **simmer** como en **anylogic**. Las corridas se realizaron de la siguiente forma:

Parte a:

Se ejecuto la simulación para 7, 8 y 9 ambulancias, 15 veces cada una con semillas distintas obteniendo los siguientes resultados:

tabla #2.

Número de Ambulancias	Número de veces que se soporto la demanda total
7	11
8	16
9	16
Desviación Estándar	2.89
Rango	[11,16]

La **tabla #2**, muestra que se necesitan aproximadamente 8 ambulancias para cumplir con la totalidad de la demanda. Sin embargo, vemos que para un gran número de semillas con 7 ambulancias sería suficiente para cubrir la totalidad, de modo que en este aspecto se nota como el valor de la semilla afecta considerablemente el resultado de la simulación

Parte b:

Se corrió la simulación para 1 sola ambulancia un total de 15 veces con semillas distintas, obteniendo los siguientes resultados:

tabla #3.

Número de Ambulancias	Número de veces que se soporto la demanda total
1	0

La **tabla #3**, muestra que una sola ambulancia no es suficiente para suplir la totalidad de la demanda, en concordancia con el punto anterior.

Parte c:

Se ejecuto la simulación para 1, 2, 3 y 4 ambulancias, 15 veces cada una con semillas distintas obteniendo los siguientes resultados:

tabla #4.

	Número de Ambulancias	Tiempo de espera promedio	Porcentaje promedio de llamadas que deben esperar	Tiempo de servicio promedio
	1	2668.82	99.74	99.99
	2	348.95	80.98	178.98
	3	62.69	47.13	179.03
	4	2.67	27.59	179.03
Rango	[1 , 4]	[2.67 , 2668.82]	[27.59 , 99.74]	[99.99, 179.03]
Desviación	1.29	1274.35	32.54	39.51

Estándar				
----------	--	--	--	--

La **tabla #4**, muestra cómo mientras vamos incrementando la cantidad de ambulancias se reduce el tiempo de espera y el porcentaje de llamadas que deben esperar, vemos como en concordancia con los incisos pasados si se siguen agregando ambulancias a la simulación llega un momento donde el tiempo de espera promedio se hace 0.

Con el fin de comprender la simulación que se está realizando se incluye además un archivo **ambulance_one_run.R** el cual funge como referencia para entender lo que sucede en una sola corrida del sistema.

III. CONCLUSIONES Y RECOMENDACIONES

En las simulaciones por eventos discretos realizadas, concuerdan con la teoría de los procesos de colas, en logro determinar las distintas peculiaridades y estadísticas que surgen cuando en un sistema de colas existen dos servidores los cuales pueden servir a un mismo cliente con el fin de agilizar el tiempo de servicio del mismo, así como también cuando existe más de un servidor y hay un número grande de clientes que entran al sistema, en este caso se determinó un aproximado de la cantidad de servidores que se deben emplear para que 500 clientes sigan durante el proceso sin que ocurra tiempo de espera y ver las diferencias en las estadísticas tomadas para distintas cantidades de clientes en el sistema.

Del mismo modo y con el fin de aportar, recomendamos que siempre que se haga este tipo de análisis utilizando los lenguajes y softwares mencionados anteriormente se revise la documentación que los grupos y empresas proveedoras de las herramientas tienen a su disposición. También se recomienda que, en caso de estar estancado en un punto, se realicen posts en los distintos grupos online que existen para las herramientas.