Práctica 3 SS: Modelos de Simulación Dinámicos y Discretos

José Manuel Pérez Lendínez, 26051613-l

December 27, 2019

Contents

1 Método de incremento fijo de tiempo.

En este caso vamos a realizar un programa que simule la carga de trabajo de un servidor. Los clientes llegan al servidor y esperan en una cola FIFO hasta que este quede libre para ser utilizado. Tanto el momento en el que llegan los clientes como el tiempo que necesitan utilizar el servidor vienen dado por dos generadores de variables aleatorias independientes e idénticamente distribuidas. Estos generadores nos devolverán un valor que determinara un momento de tiempo que pasara hasta la llegada de un nuevo cliente o el tiempo de utilización del servidor. De esta manera le sumaremos este valor el tiempo actual de nuestro simulador para obtener los datos que nos interesa.

El programa recibe 4 valores como entrada con el siguiente formado.

./simulador TiempoDeLlegada N°Clientes TiempoDeServicio N°DeRepeticiones

Las variables indicaran lo siguiente:

1. **Tiempo de llegada (tlleg)**: Sera el valor utilizado para generar los tiempos de llegadas al servidor. De forma que si le pasamos un 1 equivaldría a 60 minutos y en caso de pasarle por ejemplo 0.15 serian 9 minutos.

2. **Nº de clientes**: El programa para la ejecucion cuando atienda a un número exacto de clientes, que indicaremos con este parámetro.

3. **Tiempo de servicio(tserv)**: Utiliza los mismos tipos de valores que la variable para tiempos de llegada. Pero en este caso indicaremos el tiempo que un cliente utiliza el servidor.

4. **Nº de repeticiones**: La simulación la repetiremos un cierto número de veces para obtener la media de los resultados obtenidos. Ese número de simulaciones sera indicado por este parámetro.

El simulador nos devolverá una media de tiempo ocioso del servidor y la media de clientes en cola para utilizar el servidor.

El tiempo en la simulación incrementara de forma fija en una unidad. Vamos a realizar realizar varias mediciones y mostrarlas en la siguiente tabla para ver como se comporta el programa para valores distintos en los parámetros de tiempo de servicio y llegada.

Para estos datos he utilizada 10000 clientes y lo he repetido 50 veces para sacar las medias. Para la tabla vamos a utilizar las medidas nombradas en la práctica (horas, medias horas, minutos, segundos,décimas de segundos y milisegundos).

Para trabar con horas tlleg = 0.15 y tserv = 0.1, si se quiera por ejemplo trabajar con medias horas solo tendríamos que multiplicar estos por 2, tlleg = 0.30 y tserv = 0.2. Otro ejemplo seria trabajar con segundos que tendríamos que multiplicar 0,15 y 0.1 por 3600.

T. de llegada (tlleg)	**T. servicio(tserv)**	**Media de clientes en cola**	**Media % de tiempo ocioso Servidor**	**T. medio de ejecución (seg)**
0.15	0.1	0.0233562	0.135782	0.00116736
0.30	0.2	0.215516	2.98828	0.00159546
9	6	1.26906	31.3571	0.00200088
540	360	1.34331	33.246	0.0173753
5400	3600	1.30638	33.5554	0.150701
54000	36000	1.35409	33.237	1.37355

En la tabla se ve claramente como los valores para la media de cliente en cola y tiempo ocioso del servidor llega un momento en que prácticamente y pasan de ser muy bajas a ser siempre parecida. Esto se debe a que cunado se trabajan con unidades de tiempo altos como pueden ser las horas (tlleg = 0.15 y tserver = 0.1) o medias horas (tlleg = 0.3 y tserver = 0.2) los generadores obtienen valores muy bajos, siendo casi 0. Esto hace devuelva 1 casi siempre para el tiempo de servicio y llegada, los clientes están llegando continuamente y el servidor este ocupado prácticamente siempre. En cambio al empezar a utilizar tiempos mas pequeños, a partir de minutos (tlleg = 9 y tserver = 6), si se tiene mas en cuenta la aleatoriedad y deja de devolver solo valores próximos a 0 consiguiendo ya resultados mas realistas.

Una vez visto esto y sabiendo que con medidas de tiempo mas pequeñas(tlleg y tserv mas grandes) es mas fiable, analizaremos en estos el tiempo medio de cola y ocioso del servidor. Con esto vemos claramente como el servidor se mantiene ocioso prácticamente un 33% del tiempo y en cola tenemos una media de 1.3 clientes.

A la hora de analizar el tiempo se observa como va creciendo, esto se debe a que con unidades de tiempo mas pequeñas hace falta incrementar mas veces el reloj del simulador para llegar a un evento.

2 Método de incremento variable de tiempo.

El método de incremento variable se basa en el código anterior añadiendo dos cambios sencillos. Por esto la forma del llamar al programa es la misma que en el caso anterior.

Los cambios principales se basa en que dejamos de trabajar con el tiempo como entero y utilizaremos un float. Nuestro generadores anteriores tenían que redondear el float para devolver un número entero, en este método esto no es necesario por lo que el generador devolverla el float sin la necesidad de redondear ni comprobar si el redondeo lo convirtió en un 0. El ultimo cambio es que el reloj no aumenta de unidad en unidad, el reloj en este caso avanza hasta el evento mas cercano en el tiempo. En nuestro caso sera o la llegada de un nuevo cliente al servidor o que un cliente deje de utilizar el servido.

Vamos a ejecutar con los mismos datos que en el ejemplo anterior para comparar.

T. de llegada (tlleg)	**T. servicio(tserv)**	**Media de clientes en cola**	**Media % de tiempo ocioso Servidor**	**T. medio de ejecución (seg)**
0.15	0.1	1.32145	33.396	0.0009926
0.30	0.2	1.34731	33.1729	0.00103374
9	6	1.33189	33.3837	0.00105938
540	360	1.36657	33.4256	0.00105644
5400	3600	1.33417	33.0666	0.0010442
54000	36000	1.32711	33.2303	0.00104676

Empecemos analizando el tiempo. En este caso el tiempo no aumenta solo una unidad, sino que salta al evento mas cercano. Esto hace que el tiempo de ejecucion no dependa de la distancia entre un evento y el estado actual, si no del número de eventos que se den durante la ejecucion del simulador. Para este ejemplo he ejecutado para 10000 cliente, y cada cliente utiliza el servidor una vez. Solo tendremos 20000 eventos y serán lo que necesitemos procesar. Da igual la unidad que utilicemos, solo se tiene en cuenta la cantidad de evento que tendremos que ejecutar. Por tanto el tiempo medio de ejecucion como se ve en la tabla es el mismo para todos los ejemplos quitando las pequeñas sobrecargas que pudiera tener el procesador al ejecutarlos.

Si nos centramos en las medias de clientes en cola y tiempo ocioso del servidor, vemos que no ocurre lo mismo que con el incremento fijo. En este caso no pasamos de unas medias muy bajas a unas mas altas, sino que se mantienen las medias durante todas las ejecuciones en los mismos valores. Esto se debe a que en este caso como explique anteriormente al estar trabajando con float no tenemos el problema con unidades de tiempo grandes, como por ejemplo horas (tlleg = 0.15 y tserv = 0.1). Aunque obtengamos valores próximos a 0 con los generadores, nuestro simulador si puede trabajar con estos y no los redondea a 1 como en el apartado anterior. Con esto podemos asegurar que con el incremento variables, la unidad de tiempo que elijamos no nos afectara a la hora de obtener las medias.

Vamos a comparar nuestros resultados con los obtenidos teóricamente. Cuando $n \to +\infty$, sabes que:

$$\rho = \frac{tserv}{tIIeg},\ Q(n) \to \frac{\rho^2}{1-\rho},\ PTO(n) \to 100 * (1-\rho)$$

Vamos a sustituir y resolver con los valores para horas:

$$\rho = \frac{0.1}{0.15} = 0.667$$

$$Q(n) \to \frac{0.667^2}{1-0.667} = 1.334$$

$$PTO(n) \to 100 * (1-0.667) = 33,334$$

Con esto vemos como teóricamente tenemos resultados prácticamente iguales a los conseguidos con el incremento variable. En cambio si miramos los resultados que conseguimos con el incremento fijo:

T. de llegada (tlleg)	**T. servicio(tserv)**	**Media de clientes en cola**	**Media % de tiempo ocioso Servidor**	**T. medio de ejecución (seg)**
0.15	0.1	0.0233562	0.135782	0.00116736

Vemos que no se acercan ni lo mas mínimo a los resultados teóricos. Esto ya nos asegura todo los indicios anteriores de que el método fijo tiene problemas para conseguir resultados realistas con valores altos de tiempo.

3 Estructura de un programa de simulación dinámico y discreto.

3.1 Simulación con un único valor

En este apartado trabajaremos con un modelo mas avanzado del programa anterior. En este caso tendremos la opción de elegir el número de servidores(parámetro m) que tendremos trabajando en paralelo para atender clientes. Para comprobar que nuestro modelo trabaja correctamente, utilizaremos el caso en m=1, un tlleg = 0.15 y tserv = 0.1. Para este modelo tendremos en cuenta las siguientes medidas:

1. **Tiempo medio de espera en cola:** $\frac{tserv^2}{tlleg-tserv} = 0.2$
2. **Tiempo medio de estancia en el sistema:** $\frac{tserv*tlleg}{tlleg-tserv} = 0.3$
3. **Número medio de clientes en cola:** $\frac{tserv^2}{tlleg*(tlleg-tserv)} = 1.333$
4. **Número medio de clientes en el sistema:** $\frac{tserv}{tlleg-tserv} = 2$
5. **Longitud media de colas no vacías:** $\frac{tlleg}{tlleg-tserv} = 3$
6. **Porcentaje de tiempo de ocio del servidor:** $(1-\frac{tserv}{tlleg})*100 = 33.333$

Vamos a mostrar ahora los datos obtenidos por el simulador para estos tlleg y tserv. El formato de la llamada es el siguiente:

$$./colammk < numeroServidores >< tiempoParada >< tlleg >< tserv >$$

Para los ejemplos elegiremos los siguientes parámetros utilizando varios tiempos de parada(100,1000,10000 y 100000).

$$./colammk\ 1 < tiempoParada > 0.15\ 0.1$$

T. Parada	100	1000	10000	100000
T.medio de espera en cola	0.286	0.206	0.196	0.200
T. medio de estancia en el sistema	0.386	0.306	0.296	0.300
Num. medio clientes en cola	1.939	1.353	1.305	1.335
Num. medio clientes en sistema	2.659	2.002	1.969	2.000
Long. media de colas no vaciás	3.750	3.175	2.976	3.018
% de tiempo de ocio del servidor	27.983	35.171	33.569	33.518

Si comparamos los valores obtenidos y los valores teóricos para nuestras salidas, vemos claramente como van acercándose todos los para metros a estos valores conforme subimos el tiempo de parada. Llegado a un tiempo de parada igual a 10^5, vemos como ya obtenemos parámetros prácticamente iguales a los teóricos. Esto nos dice que partir de 10^5 obtendremos valores muy aproximados a los teóricos y que nuestras simulaciones tendrían que acercarse a este número para obtener resultados que podamos dar como buenos.

3.2 Simulación con varios servidores.

En este apartado vamos a analizar como afecta a nuestros resultados tener un mayor número de servidores, aunque aumentaremos el tserv para los servidores. De esta forma intentaremos igualar las condiciones de un único servidor mas rápido y varios servidores mas lentos para ver como afecta el aumento en número de servidores bajo condiciones parecidas. Para igualar el tiempo de servidor, si para un único servidor usamos tserv, para m servidores utilizamos un tiempo igual m*tserv. Realizaremos las pruebas con tiempo de parada 10^5 y aumentaremos el número de servidores de 2 en 2.

Num. Servidores	1	3	5	7	9	11
tserv	0.1	0.3	0.5	0.7	0.9	1.1
T.medio de espera en cola	0.200	0.132	0.099	0.076	0.057	0.045
T. medio de estancia en el sistema	0.300	0.432	0.599	0.776	0.957	1.145
Num. medio clientes en cola	1.335	0.881	0.661	0.506	0.382	0.302
Num. medio clientes en sistema	2.000	2.881	3.991	5.182	6.382	7.644
Long. media de colas no vaciás	3.018	2.987	3.027	3.013	2.943	2.918
% de tiempo de ocio del servidor	33.518	33.341	33.392	33.206	33.334	33.261

Vamos a analizar estos datos detenidamente, para determinar el comportamiento del simulador y que el programa funciona correctamente.

1. **T. medio de espera en cola:** En este caso se ve claramente una mejora. A mas servidores menos esperaran los clientes en cola. Esto se debe a que aunque se suba el tiempo que tarde un servidor en ejecutar la tarea(tserv), tendremos mas servidores a los que dar tareas y al final en vez de estar en cola esperando, estarán en un servidor ejecutando.

2. **T. medio de estancia en el sistema:** Este parámetro es todo lo contrario al anterior. Si los clientes en vez de en cola están en el servidor y el tserv aumenta también, se tendrá un aumento en este parámetro cada vez que aumente el número de servidor y el tserv.

3. **Num. medio de clientes en cola:**Este parámetro también cae conforme mas servidores tenemos. Esto se da por lo explicado anteriormente también. Tenemos mas servidores y los clientes que llegan es mas probable que se asignen directamente a un servidor o tengan pocos clientes en la cola antes que el.

4. **Número medio de clientes en sistema:** En este caso al tener un aumento en el tserv este parámetro crecerá, estarán mas tiempo en el sistema al necesitar mas tiempo del servidor para ejecutar su tarea.

5. **Porcentaje de tiempo de ocio del servidor:** El porcentaje de ocio no varia aunque se aumente los servidores. Esto se da porque aunque aumentemos los servidores también se aumenta el tiempo de servicio, por tanto se mantiene el tiempo de ocio igual.

Como conclusión de aumentar los servidores, tendremos una mejora para los clientes a la hora de esperar para poder realizar su servicio, aunque también tendremos un mayor tiempo de uso de los servidores y el gasto que esto puede conllevar, aunque esto se da porque se aumenta también el tserv. No obtendremos cambios en el tiempo de ocio del servidor, por lo que tendremos un 33% de tiempo que los servidores no estarán realizando tarea.

3.3 Cambios en el software.

3.3.1 Añadir medias y derivación típica.

Vamos a realizar un cambio en el simulador para realizar varias ejecuciones y obtener las medias y desviación típica para estas medias.
Para analizar este cambio vamos a ver como afectan el número de simulaciones probando 10,100 y 1000 simulaciones, para el programa con 1 servidor, tserv = 9, tlleg = 6 y un tiempo de parada de 10000. Cambio este tsev y tlleg a estos valores porque al trabajar con estos números, el tiempo de ejecución es mas rápido y puedo realizar mas simulaciones.

Para la ejecucion solo tenemos que añadir un nuevo parámetro que nos da el número de simulaciones:

$./colammk < numeroServidores >< tiempoParada >< tlleg >< tserv >< numRepeticiones >$
Vamos a mostrar los datos medidos.

Num. Simulaciones	100	1000	2000
T.medio de espera en cola	12.002 ± 2.473	11.862 ± 2.390	11.854 ± 2.475
T. medio de estancia en el sistema	18.002 ± 2.473	17.862 ± 2.390	17.854 ± 2.475
Num. medio clientes en cola	1.338 ± 0.286	1.325 ± 0.287	1.320 ± 0.295
Num. medio clientes en sistema	2.002 ± 1.529	1.991 ± 1.519	1.985 ± 1.516
Long. media de colas no vaciás	3.006 ± 0.486	2.962 ± 0.445	2.965 ± 0.467
% de tiempo de ocio del servidor	33.537 ± 2.686	33.321 ± 2.687	33.534 ± 2.801
Longitud de la máxima cola	12.705 ± 3.155	12.200 ± 2.486	12.991 ± 3.210

Como se ve en la tabla, tenemos una desviación muy parecida entre unos casos y otros. La diferencia principal es que por ejemplo cuando realizamos solo 100 simulaciones los resultados que nos daría pueden variar mas que en los demás. Para probar esto vamos a ejecutar otras dos veces con 100 simulaciones y vamos a comparar los datos.

Num. simulacion	1	2	3
T.medio de espera en cola	12.002 ± 2.473	11.415 ± 2.286	12.797 ± 3.322
T. medio de estancia en el sistema	18.002 ± 2.473	17.415 ± 2.286	18.799 ± 3.322
Num. medio clientes en cola	1.338 ± 0.286	1.267 ± 0.272	1.434 ± 0.401
Num. medio clientes en sistema	2.002 ± 1.529	1.928 ± 1.490	2.107 ± 1.607
Long. media de colas no vaciás	3.006 ± 0.486	2.886 ± 0.446	3.129 ± 0.622
% de tiempo de ocio del servidor	33.537 ± 2.686	33.831 ± 2.694	32.727 ± 2.880
Longitud de la máxima cola	12.705 ± 3.155	12.350 ± 2.750	13.480 ± 3.454

Como se ve en este caso si tenemos desviaciones y valores distintos para las tres simulaciones con los mismos parámetros. En cambio cuando tenemos un número de simulaciones mayor, estas desviaciones no tienen grandes variaciones. Realizaremos una segunda ejecucion con 1000 simulaciones para mostrarlo.

Num. simulacion	1	2
T.medio de espera en cola	11.415 ± 2.286	11.824 ± 2.553
T. medio de estancia en el sistema	17.415 ± 2.286	17.824 ± 2.553
Num. medio clientes en cola	1.267 ± 0.272	1.317 ± 0.301
Num. medio clientes en sistema	1.928 ± 1.490	1.980 ± 1.514
Long. media de colas no vaciás	2.886 ± 0.446	2.968 ± 0.484
% de tiempo de ocio del servidor	33.831 ± 2.694	33.663 ± 2.723
Longitud de la máxima cola	12.350 ± 2.750	12.999 ± 3.189

Como se ve en este caso tenemos unos valores medios y desviaciones mas próximos unos a otros y con menores cambios.

3.3.2 Añadir distintos generadores

En este caso vamos a añadir dos tipos de generadores distintos para compararlos. Añadiremos un generar uniforme y otro generador que siempre devuelva el valor medio.Nuestro generador aleatorio devolverá un valor entre 0 y $2 * tser$ o $2 * ttleg$ dependiendo de lo que necesitemos. Todos los valores tienen la misma probabilidad de tocar. En cambio el generador de media siempre nos devuelve el mismo valor, que corresponde con la el valor de tserv o tlleg que pasemos.

Vamos a realizar la prueba para los tres generadores que tenemos, con los siguientes parámetros:

1. **Número de servidores:** 1
2. **Tiempo de parada:** 10000
3. **Tlleg:** 9
4. **Tserv:** 6
5. **Número de repeticiones:** 1000

Vamos a mostrar en la siguiente tabla los resultados de ejecutar estros tres generadores.

Tipo de generador	**Exponencial**	**Uniforme**	**Medio**
T.medio de espera en cola	11.415 ± 2.286	3.584 ± 0.525	0 ± 0
T. medio de estancia en el sistema	17.415 ± 2.286	9.584 ± 0.525	6 ± 0
Num. medio clientes en cola	1.267 ± 0.272	0.399 ± 0.063	0 ± 0
Num. medio clientes en sistema	1.928 ± 1.490	1.065 ± 0.991	0.6 ± 0.6
Long. media de colas no vaciás	2.886 ± 0.446	1.499 ± 0.124	0 ± 0
% de tiempo de ocio del servidor	33.831 ± 2.694	33.408 ± 1.656	33.390 ± 0
Longitud de la máxima cola	12.350 ± 2.750	5.686 ± 1.220	0 ± 0

Se ve como ninguno de estos dos nuevos generadores se acercan a los valores teóricos que se obtuvieron en los apartados anteriores y el exponencial si los obtiene. Esto nos indica que no son buenos generadores para este problema, mostrándonos la importancia que tiene elegir un buen generador que se adapte a nuestro problema.

Dentro de los dos nuevos el mejor es el uniforme ya que el medio directamente nos devuelve todos los valores prácticamente 0, lo que nos indica que la llegada de clientes para este generador hace que lleguen los clientes siempre después de que el cliente anterior ya salio de nuestro sistema.

Los únicos valores que si se aproxima en los tres generados a los teóricos, es el porcentaje de tiempo de ocio del servidor, esto es debido a que atienden a un número de clientes prácticamente igual y por tanto tendrán también la una cantidad tareas a realizar parecida, solo variando el tiempo de la tarea, que hace que la desviación si sea distinta para los tres caso.

4 Remolcador de un puerto.

4.1 Prestaciones del sistema.

Para probar los resultados vamos a realizar distintas ejecuciones con 10,100 y 1000 simulaciones. Vamos a comparar estos resultados en la siguiente tabla:

Num. simulaciones	**10**	**100**	**1000**
Num. medio barcos en cola de llegada	1.1019 ± 0.3485	1.1340 ± 0.3843	1.2274 ± 0.5185
Num. medio barcos en cola de salidas	0.0265 ± 0.0041	0.0287 ± 0.0031	0.0290 ± 0.0034
Tiempo medio estancia en puerto(t 0)	32.5089 ± 3.6048	33.0585 ± 7.2307	33.8608 ± 5.5048
Tiempo medio estancia en puerto(t 1)	38.2785 ± 4.0889	38.8529 ± 4.0461	39.8520 ± 5.5709
Tiempo medio estancia en puerto(t 2)	50.3931 ± 3.7226	50.5921 ± 4.2212	51.7220 ± 5.7109
% tiempo remolcador desocupado	80.5595 ± 0.1545	80.6229 ± 0.1628	80.6299 ± 0.1963
% tiempo remolcador viajando vació	1.3482 ± 0.3286	1.2711 ± 0.2389	1.2563 ± 0.2286
% tiempo remolcador llevando barco	18.0923 ± 0.2671	18.1060 ± 0.2389	18.1138 ± 0.2382
% tiempo de puntos de atraque libre	13.5710 ± 2.3110	13.0120 ± 1.4359	12.9967 ± 1.3794
% tiempo de puntos de atraque ocupadas sin cargar	0.8827 ± 0.1354	0.9557 ± 0.1019	0.9671 ± 0.1131
% tiempo de puntos de atraque ocupadas cargando	85.5464 ± 2.2571	86.0323 ± 1.4156	86.0362 ± 1.3645

Como se ve en las diferencias entre 100 y 1000 simulaciones es muy pequeña en casi todas las variables que calcula el programa, en cambio si tenemos una diferencia mayor con 10 simulaciones. Por tanto a partir de ahora usaremos 1000 simulaciones para las siguientes ejecuciones que realicemos.

Empezando por las primeras variable se ve claramente como se suele tener en cola de llegada un solo barco en espera para ser cargado, todo lo contrario ocurre cuando se quiere abandonar el puerto por parte de un barco. Esta variable esta muy cercana a 0, por tanto, no tendremos barcos en espera para salir del puerto, saliendo justo después de cargar prácticamente.

El tiempo de estancia medio en puerto, tiende a ser unas 14 o 15 horas mas que el tiempo de carga. Esto se cumple en los tres tipos de barco, lo que nos indicaría que aparte de el tiempo de carga, en las demás tareas (llegar al anclaje, salir del puerto y esperas para el remolcador) se tardaría estas 14 o 15 horas extra.

En cuanto al remolcador se ve que esta un 80% del tiempo desocupada, esto podría ser debido a que los barcos tardan mucho en ser cargados y en ese tiempo el remolcador solo esta a la espera de una nueva llegada. El resto del tiempo el remolcador utiliza un 18% en remolcar barcos y solo un 1.2% en los viajes realizados para llegar a los barcos en espera para entrar o volver al puerto sin remolcar nada.

El tiempo en punto de atraque ocupados sin carga nos demuestra lo dicho anteriormente, ya que solo es un 0.95% del tiempo y esto nos indicaría que los barcos no esperan mucho a ser llevados fuera del puerto despumes de ser cargados. Los puntos de atraque pertenecen un 86% del tiempo ocupado.

4.2 Modificaciones del simulador.

En este caso vamos a realizar y analizar las siguientes modificaciones:

1. **M1:** Modificar el número de puntos de atraque a 4.
2. **M2:** Modificar el número de puntos de atraque a 5.
3. **M3:** Añadir remolcador sin que le afecte las tormentas.
4. **M4:** Añadir un remolcador mas rápido, de 0.25 a 0.15.

Para estos análisis añadiremos una nueva variable que calculara el total de toneladas cargadas. Vamos a mostrar los resultados.

opcion	no modf	M1	M2	M3	M4
Num. medio barcos en cola de llegada	1.2274	0.0866	0.0473	1.0183	1.2033
Num. medio barcos en cola de salidas	0.0290	0.0289	0.0293	0.0109	0.0281
Tiempo medio estancia en puerto(t 0)	33.8608	21.2781	20.8495	31.3566	33.6767
Tiempo medio estancia en puerto(t 1)	39.8520	27.2861	26.8474	37.3550	39.5179
Tiempo medio estancia en puerto(t 2)	51.7220	39.2557	38.8356	49.2504	51.4262
% tiempo remolcador desocupado	80.6299	78.2378	77.8616	80.5312	81.1309
% tiempo remolcador viajando vació	1.2563	3.6381	3.9926	1.3677	0.7665
% tiempo remolcador llevando barco	18.1138	18.1241	18.1459	18.1011	18.1027
% tiempo de puntos de atraque libre	12.9967	34.7102	47.6846	13.6251	13.0614
% tiempo de puntos de atraque ocupadas sin cargar	0.9671	0.7213	0.5856	0.3644	0.9373
% tiempo de puntos de atraque ocupadas cargando	86.0362	64.5686	51.7298	85.9874	86.0012
Media de toneladas cargadas	1780254	1783296	1785603	1781629	1781032

La primera columna muestra los datos sin realizar ninguna modificación. Una vez sabido esto pasemos a analizar los resultados.

Vamos a analizar primero las modificaciones de aumento par puntos de atraque(M1 y M2). En este caso se ve claramente como el aumento de atraques mejora siempre la cantidad de barcos en cola tanto para salir como para entrar del puerto. Se mejora los tiempos de espera al tener mas puntos de atraque reduciendo así el tiempo medio de estancia en el puerto al esperar menos tiempo para entrar. El tiempo de remolcador desocupado mejora un poco, aunque no mas de un 3%, aumentando el porcentaje de tiempo viajando vació. También cambiaran el porcentaje de tiempo que los puntos de atraque están ocupado cargando, puesto que al tener mas atraques, tendremos mas posibilidad de que un punto de atraque este vació.

La modificación para que al remolcador no le afecte las tormentas mejora un poco el número medio de barcos en cola de llegada y salida, aunque menos que en el caso anterior. Los tiempos de estancia en puerto también mejoran un poco pero también menos que antes. Esto se debe a que no tendrán que esperar para entrar o salir a que se termine una tormenta. El tiempo de puntos de atraque sin cargar también mejora por el mismo motivo.

Si se modifica la velocidad del cargero(M4) se consiguen mejorar en todos los datos que tiene que ver con los remolcadores(espera en colas, tiempos de estancia, tiempos de remolcador y demas) pero siendo una mejora muy pequeña, puesto que los remolcadores tienen mucho tiempo desocupado y esto hace que no afecte mucho al sistema. El único parámetro que si mejora mas es el tiempo viajando vació del remolcador.

En cuanto a las toneladas cargada de media para cada ejecucion, vemos como el mejor dato lo obtenemos en el caso de aumentar los números de atraques, siendo el mejor en este caso 5 atraques. Las otras dos mejoras y el valor inicial del programa sin mejora nos dan valores muy parecidos entre si. Y aunque los puntos de atraque nos dan una mejora tampoco es muy grande. Debido a todo lo analizado anteriormente, la mejor opción para mejorar el puerto seria aumentar le número de atraques que tengamos disponibles.

5 Análisis de salida y Experimentación.

5.1 Número de simulaciones.

Para este apartado vamos a analizar el número de repeticiones y simulaciones necesarias para tener buenos resultados. Vamos a quedarnos con la variable número medio de barcos en cola de atraque(NBCA). Para esto compararemos el modelo original($NBCAA$) y el modelo con remolcador al que no afectan las tormentas($NBCAB$).

Vamos a quedarnos con las veces que gana cada uno de los dos modelos, de forma que se realizaran 100 ejecuciones y con diferentes número de simulación(1,5,10,25,50). No voy a realizar la ejecucion con 100, ya que en mi caso

la ejecucion con 50 ha tardado unos 5 minutos en ejecutar. Para que tardara un poco menos solo ejecuto una vez el modelo sin mejores y estos datos son los que comparo con los dos modelos con mejora.

Para realizar esto he modificado el archivo de puerto para que solo devuelva el valor numérico que queremos analizar en este caso(puesto3.cpp). Ademas he realizado un pequeño script bash que ejecute este programa sin modificación y con las dos modificaciones.

Num Simulaciones	**1**	**5**	**10**	**25**	**50**
Simulador inicial	26%	20%	15%	11%	2%
Mejora tormentas	74%	80%	85%	89%	98%

En el caso del remolcador al que no le afectan las tormentas se ve claramente como al subir el número de simulaciones, obtenemos un resultado muy superiores pasando de un 74% con menos simulaciones, a un 98% con 100 simulaciones. Esto nos indica que los datos tomados con pocas simulaciones no son muy fiables, puesto que tenemos una gran diferencia con respecto a los de mayores simulaciones.

Num Simulaciones	**1**	**5**	**10**	**25**	**50**
Simulador inicial	39%	43%	54%	42%	45%
Mejora velocidad	61%	57%	46%	58%	55%

En el caso de mejorar el aumento de velocidad, se ve que están bastante mas igualados aunque en 4 de las 5 simulaciones ha ganado la mejora. En este ejemplo se ve como en el ejemplo con 1 simulación nos dan resultados que nos podrían hacer creer que el la mejora es mucho mejor de lo que es realmente. Llegando a estabilizarse al rededor de un 5%-8% de mejora en los casos que tienen mas simulaciones.

Como conclusión tenemos que el número de simulaciones es muy importante en este método para obtener valores realistas y que no fallemos al decantarnos por uno u otro modelo. Siendo mas difícil en casos en los que la mejora no es muy clara.

5.2 Intervalo de confianza.

Vamos a realizar los intervalos de confianza mediante un script de python para procesar los datos y un script bash para obtenerlos y almacenar. Para este caso ejecutaremos el simulador 120 veces, simulando en cada ocasion 20 veces. Después de obtener los datos calcularemos la resta de el simulador al que no afecta la tormenta y el simulador con remolcador mas rapido. Por ultimo obtendremos la media y la varianza para poder obtener al final el intervalo de confianza. En nuestro caso nuestro la distribución-t de Student(implementada con una libreria de python) con un intervalo de confianza del 95%.

Nºsimulaciones	**Intervalo**	**Media**	**Varianza**
10	[-0.00419, 0.22730]	0.11155	0.11155
25	[-0.00252, 0.14217]	0.06982	0.03072
50	[0.07572, 0.17852]	0.12712	0.03270
75	[0.10114, 0.20249]	0.1518	0.04850
100	[0.12456, 0.20871]	0.16663	0.04496
125	[0.14947, 0.22494]	0.18721	0.04543

Pasemos a analizar los datos, como se ve en los intervalos de poscas simulaciones tenemos el 0 entre ambos extremos. Esto nos indica que en esos rangos de repeticiones ninguno de los simuladores es muy superior. En cambio al ir aumentado el número de simulaciones, vemos como el 0 no aparece en los intervalos. Al estar los resultados en el lado positivo del intervalo podemos decir que la mejor opción para nuestro simulador es la modificación que se encuentra a la derecha en nuestra resta para obtener los datos. En este caso al utilizar $Media\ mejora\ Velocidad - Media\ sin\ tormenta$, podemos saber que la mejora que mayor beneficios nos da es que no afecte a los remolcadores las tormentas.

5.3 Comparación de varios sistemas.

En este caso vamos a comprar los dos sistemas del apartado anterior y añadiremos también un modelo con 4 y 5 puntos de atraque. Utilizaremos 40 muestras para este ejemplo(cada una sera una ejecucion con 40 simulaciones). Ademas queremos tener una buena probabilidad(P^*), por tanto usaremos 95%. Consultando la tabla para el parámetro h, vemos que para un modelo con k=4 y P^*=95% tenemos que elegir un h=3.003. Ya solo nos queda elegir el parámetro $d*$, como los valores en la variable que nosotros vamos a estudiar(número de barcos en cola de llegada), por tanto este valor lo marcaremos $d*$=0.1 y indica que las diferencias en el caso de ser menor que este parámetro, es muy pequeña y no es significativa. Con esto tenemos elegidos ya todos los parámetros necesarios.

Mejora	$X_i^{(1)}(40)$	$s_i^2(40)$	N_i	$X_i^{(2)}(N_i - 40)$	W_{i1}	W_{i2}	$X_i^p(N_i)$
tormenta	1.0049	0.0038	41	0.9529	1.4851	-0.4851	1.0302
velocidad	1.2461	0.0139	41	1.2316	1.2080	-0.2080	1.2491
4atraques	0.0867	$6.588 * 10^{-6}$	41	0.0856	13.7886	-12.7886	0.1017
5atraques	0.0470	$3.803 * 10^{-7}$	41	0.0472	54.3109	-53.3109	0.04069

Para obtener el mejor simulador tendremos que quedarnos con el que tenga un valor menor para $X_i^p(N_i)$. En este caso el menor valor nos lo da la mejora con 5 atraques, por tanto podríamos decir que esta es la mejor opción, aunque el modelo con 4 atraques no tiene grandes diferencias con el anterior.