

Simulación de Sistemas (2019-2020)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Práctica 2



**UNIVERSIDAD
DE GRANADA**

Antonio Jesús Heredia Castillo

1 de noviembre de 2019

Índice

1. Mi Segundo Modelo de Simulación de Monte Carlo	4
1.1. Modelización por Monte Carlo	4
1.2. Modificaciones del Modelo	7
1.2.1. Coste de devolución fijo	7
1.2.2. Coste de devolución fijo pero decidiendo si devolver o no	9
2. Generadores de datos	11
2.1. Mejorando los Generadores	11
2.1.1. Reordenar valores	11
2.1.2. Búsqueda binaria	12
2.1.3. Generación en tiempo constante	12
2.1.4. Comparación de los distintos metodos.	12
2.2. Generadores congruenciales	14
3. Fe de erratas	14

Índice de figuras

1.	Diferencias entre la cantidad de simulaciones.	6
2.	Diferencias entre los costes de devolución	7
3.	Ganancias con coste fijos.	9
4.	Ganancias con coste fijos.	9
5.	Ganancias con costes fijos decidiendo si devolver o no.	11
6.	Ganancias con costes fijos decidiendo si devolver o no.	11
7.	Diferencias de tiempo entre las distintas mejoras.	13
8.	Diferencias de tiempo entre las distintas mejoras.	13

1. Mi Segundo Modelo de Simulación de Monte Carlo

1.1. Modelización por Monte Carlo

Para probar los resultados practicos de la modelización por Monte Carlo he realizado una serie de experimentos con distintos valores (los especificados en el gui3n de la practica). Estos valores ser3:

1. Ganancia: 10 Perdida: 1
2. Ganancia: 10 Perdida: 5
3. Ganancia: 10 Perdida:10

Para ver como afecta la simulaci3n de mas o menos d3as, los experimentos anteriores los simularemos con 100,1000,5000,10000 y 100000 d3as. Al usar n3meros aleatorios podemos presuponer que cuantas mas d3as simulemos tendremos una visi3n mas clara de lo que podr3 ocurrir.

En la siguiente tabla podemos ver los datos generados usando una distribuci3n uniforme.

Ganancia	Perdida	Repeticiones	Mejor pedido	Mejor ganancia
10	1	100	100	483.769989
10	1	1000	77	460.899994
10	1	5000	90	453.419800
10	1	10000	89	453.176788
10	1	100000	89	450.824799
10	5	100	68	373.850006
10	5	1000	69	345.690002
10	5	5000	61	335.872009
10	5	10000	67	332.891510
10	5	100000	70	330.063629
10	10	100	61	292.399994
10	10	1000	51	249.539993
10	10	5000	57	249.584000
10	10	10000	54	251.475998
10	10	100000	47	247.018402

Tabla 1: Usando el generador de datos a

En los datos recogidos en las distintas tablas podremos comparar el funcionamiento de los distintos generadores de datos. Una cosa que va a ser común en todos los generadores sera que a medida que haces mas simulaciones los precios son mas bajos. Esto tiene una sencilla explicación, y es que con pocas simulaciones podemos tener una en que de la casualidad que todos los días vence todos los periódicos pero en cambio, al realizar un numero suficiente de simulaciones tenemos que las media de periódicos vendidos se vuelve parecida.

En esta tabla podemos ver los datos obtenidos en una distribución proporcional.

Ganancia	Perdida	Repeticiones	Mejor pedido	Mejor ganancia
10	1	100	84	351.380005
10	1	1000	74	304.058990
10	1	5000	74	286.687805
10	1	10000	76	288.766602
10	1	100000	73	284.473694
10	5	100	39	227.850006
10	5	1000	48	203.309998
10	5	5000	49	192.845001
10	5	10000	44	191.050995
10	5	100000	41	188.323486
10	10	100	32	178.600006
10	10	1000	34	148.220001
10	10	5000	29	139.632004
10	10	10000	28	136.710007
10	10	100000	31	134.704605

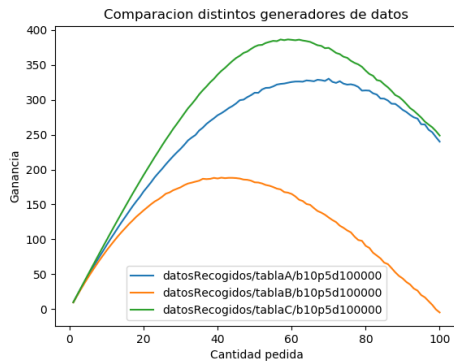
Tabla 2: Usando el generador de datos b

Sin duda la distribución proporcional es la que peores datos nos devuelve. En la siguiente tabla podemos ver los datos obtenidos a partir de la distribución triangular.

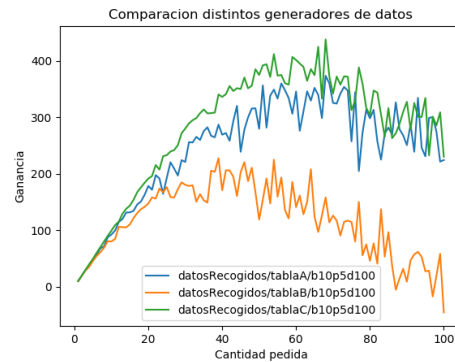
Ganancia	Perdida	Repeticiones	Mejor pedido	Mejor ganancia
10	1	100	73	529.580017
10	1	1000	79	476.423004
10	1	5000	75	470.261200
10	1	10000	77	465.843414
10	1	100000	80	465.232147
10	5	100	68	438.200012
10	5	1000	62	407.975006
10	5	5000	56	388.279999
10	5	10000	60	389.065491
10	5	100000	59	386.263916
10	10	100	61	382.000000
10	10	1000	53	342.059998
10	10	5000	48	338.799988
10	10	10000	51	337.914001
10	10	100000	50	334.353210

Tabla 3: Usando el generador de datos c

Como se puede observar en la Figura 1a, sin duda el mejor resultado lo obtenemos usando la distribución triangular. Esto puede ocurrir por un motivo y es que la distribución triangular favorece los datos que se encuentran en el centro y esto puede ser una buena estrategia ya que no va vender ni muchos ni pocos periódicos, haciendo así que las perdidas no suban demasiado respecto a las ventas.



(a) 100000 simulaciones



(b) 100 simulaciones

Figura 1: Diferencias entre la cantidad de simulaciones.

Otra cosa interesante que podemos analizar en la Figura 1 es como armoniza los resultados con muchas simulaciones. Esto se debe a que con pocas simulaciones podemos tener datos muy buenos o muy malos ya que depende mucho de los datos que se generen.

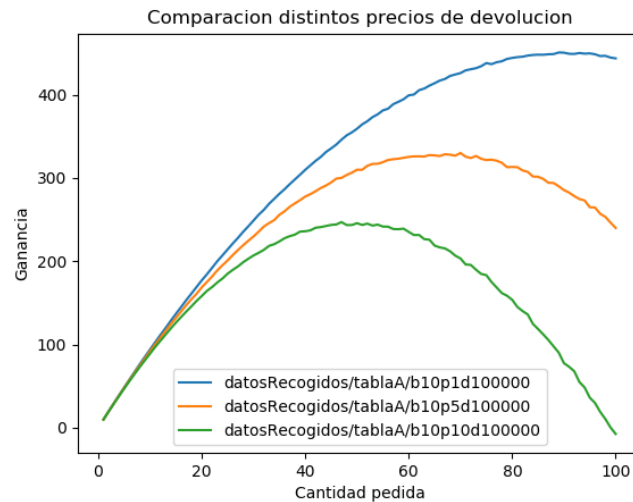


Figura 2: Diferencias entre los costes de devolución

Aunque era de esperar podemos ver en la Figura 2 como cuanto mas nos cueste devolver un periódico mas recomendable es pedir de forma controlada. Ya que si podemos muchos pero no los vendemos podemos llegar a tener perdidas.

1.2. Modificaciones del Modelo

Para esta parte de la practica vamos a trabajar sobre el modelo de Monte Carlo anterior pero añadiéndole algunas mejoras.

1.2.1. Coste de devolución fijo

La base del modelo sera el anterior pero esta vez en vez de pagar según la cantidad de periódicos que devuelvas la devolución tendrá un coste fijo. Veremos como influye esto a la hora de ver las ganancias que tendremos. Esta vez las variables serán:

1. Ganancia: 10 Coste devolución: 10
2. Ganancia: 10 Coste devolución: 50

3. Ganancia: 10 Coste devolución:100

Y como hemos visto que cuantas mas ejecuciones mas preciso sera el modelo, lo ejecutaremos 100000 veces.

Repeticiones	Ganancia	Perdida	Mejor pedido	Mejor ganancia
a	10	10	96	486.497528
a	10	50	94	447.436249
a	10	100	92	400.943359
b	10	10	98	321.195709
b	10	50	97	281.260712
b	10	100	76	231.906296
c	10	10	94	490.859680
c	10	50	91	451.618713
c	10	100	77	403.607269

Tabla 4: Modificación 1

Como podemos ver en la Tabla 4, al igual que pasaba antes el generador de tipo triangular nos da las mejores ganancias. Aunque aqui lo que podemos ver es como no influye tanto los costes fijos a la hora de tener el mejor pedido. Ya que nos importa poco cuantos devolvamos que siempre vamos a tener que pagar lo mismo. Esto se puede ver en la diferencia entre las mejores ganancias, que entre cada una los separa lo mismo que en los costes fijos por devolución.

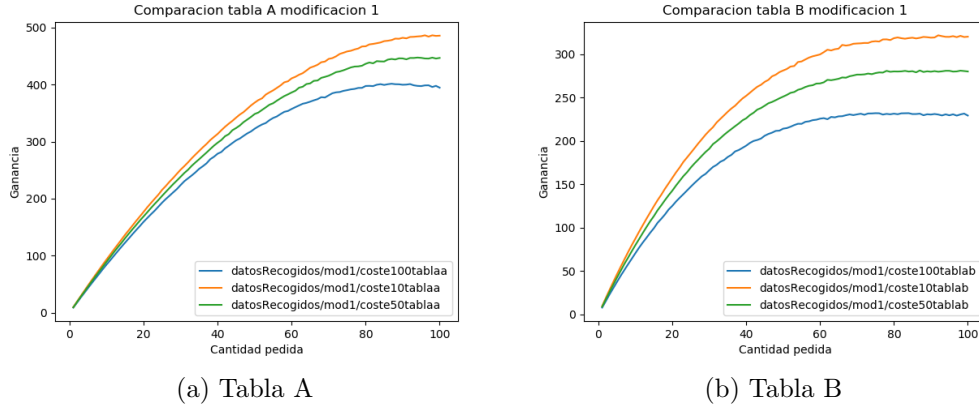


Figura 3: Ganancias con coste fijos.

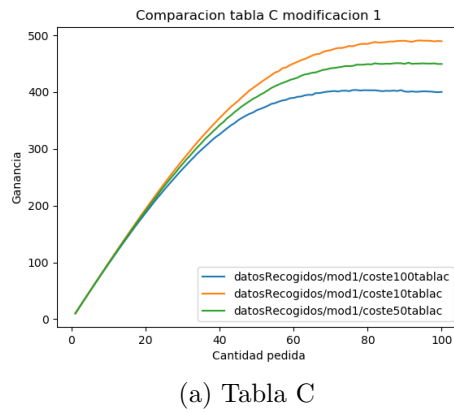


Figura 4: Ganancias con coste fijos.

Como podemos ver en la Figura 4 todas las tablas se comportan de la misma manera. Una vez llegan a una cantidad de pedidos, se comportan casi de forma constante y no crecen ni disminuyen los beneficios.

1.2.2. Coste de devolución fijo pero decidiendo si devolver o no

En esta modificación añadimos otra mejora. Esta vez además de tener un coste fijo de devolución el kiosquero podrá decidir cuando devolver o no, teniendo en cuenta cuanto perdería por periódico en caso de que no lo hiciera. Para realizar la comparación fijaremos tanto la cantidad de simulaciones como la ganancia y el coste de devolución. Este ejemplo lo ejecutaremos con las siguientes variables:

1. Ganancia: 10 Coste devolución: 100 Coste no devolución: 5

2. Ganancia: 10 Coste devolución: 100 Coste no devolución: 10
3. Ganancia: 10 Coste devolución: 100 Coste no devolución: 15

Repeticiones	Ganancia	Coste fijo	Perdida	Mejor pedido	Mejor ganancia
a	10	10	5	88	411.015045
a	10	50	10	92	406.135590
a	10	100	15	90	404.093689
b	10	10	5	72	238.302399
b	10	50	10	71	234.886505
b	10	100	15	77	233.997498
c	10	10	5	71	416.182068
c	10	50	10	73	408.292480
c	10	100	15	77	406.603485

Tabla 5: Modificación 2

Como esta vez hemos fijado los costes fijos y lo que variamos son las perdidas por no devolver lo que tenemos es que como solo devuelve cuando esta seguro de que las perdidas van a ser mayores que los costes fijos no nos importa el valor de los mismo. De esta forma podríamos aprovechar al máximo las ganancias. A diferencia que en la Figura 4 en la Figura 6 podemos ver es que aquí si importa que pidamos periódicos de mas ya que no se comporta de manera constante. Si pedimos de mas los beneficios empiezan a bajar de forma muy suave.

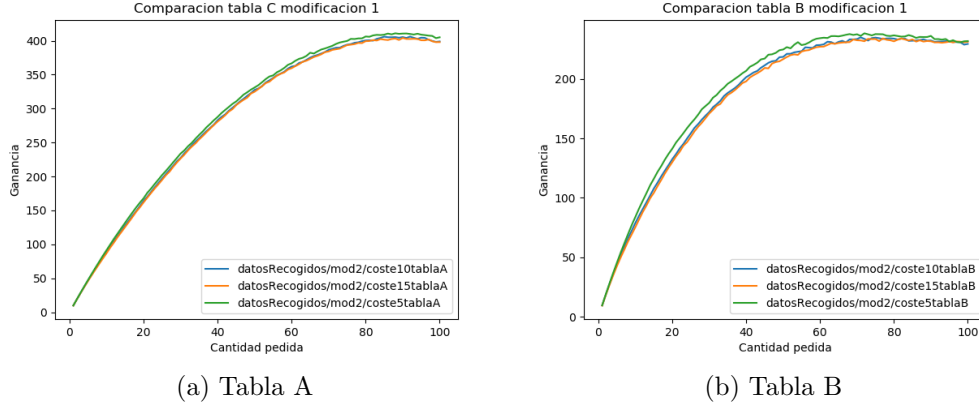


Figura 5: Ganancias con costes fijos decidiendo si devolver o no.

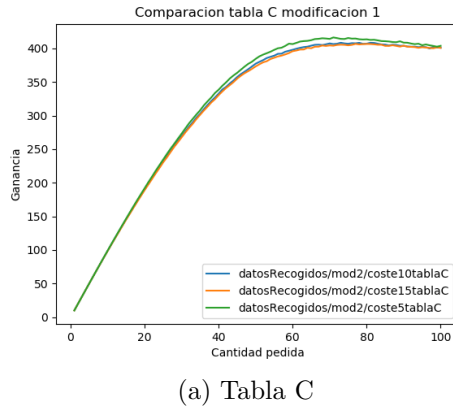


Figura 6: Ganancias con costes fijos decidiendo si devolver o no.

2. Generadores de datos

2.1. Mejorando los Generadores

Para mejorar las tablas vamos a usar varios métodos para cambiar la forma en que se generan.

2.1.1. Reordenar valores

En este caso solo tenemos que modificar el generador C, es decir el de distribución triangular. Esto se debe a que en el a los valores están ordenados puesto que son iguales en cambio y en el b si que se generan de forma decre-

ciente. Para ello hemos modificado el generador que nos daba el profesor para generarlo de forma decreciente sin tener que re ordenarlos después. Además he tenido que hacer cambios tanto en la generación de los otros dos métodos como en la función que nos devuelve un valor puesto que he usado unos pares de valores para guardar tanto el valor como su probabilidad acumulada ya que esta vez el índice donde se encuentra no corresponde con el valor que es. Para ver una comparativa de como es de eficiente cada uno los ejecutare varias veces para comprobar tiempos.

2.1.2. Búsqueda binaria

En este método la generación de las tablas no cambia, lo que cambiaremos es usar una búsqueda binaria a la hora de obtener un valor. Haciendo así una búsqueda mas rápida que nos puede venir mejor.

2.1.3. Generación en tiempo constante

Para este método no usaremos una tabla, simplemente generaremos un dato a partir de lo devuelto por la función *uniforme()* multiplicandolo por el valor maximo que queremos obtener. Esto funciona gracias a que lo que nos devuelve *uniforme()* se encuentra entre 0 y 1.

2.1.4. Comparación de los distintos métodos.

Para la comparación como en el generador 3 siempre se da los mismos datos, he usado los mismos datos recogidos para las tres comparaciones.

Como podemos ver en la Figura 7a el original como su modificación para el generador 1 son casi iguales. La pequeña diferencia que existe es por el hecho de trabajar con pares de valores. En cambio en el la obtención de datos usando búsqueda binaria si que vemos una gran mejora. El tiempo de ejecución se disminuye hasta la mitad, haciendo así una buena optimización para cuando tengamos que ejecutarlo muchas veces. Aunque sin dudar lo el que mejor funciona en este caso y en los siguientes es el generador en tiempo constante que por mucho que aumentemos las ejecuciones apenas aumenta el tiempo de ejecución. Por tanto el generador en tiempo constante siempre va a ser el mas rápido.

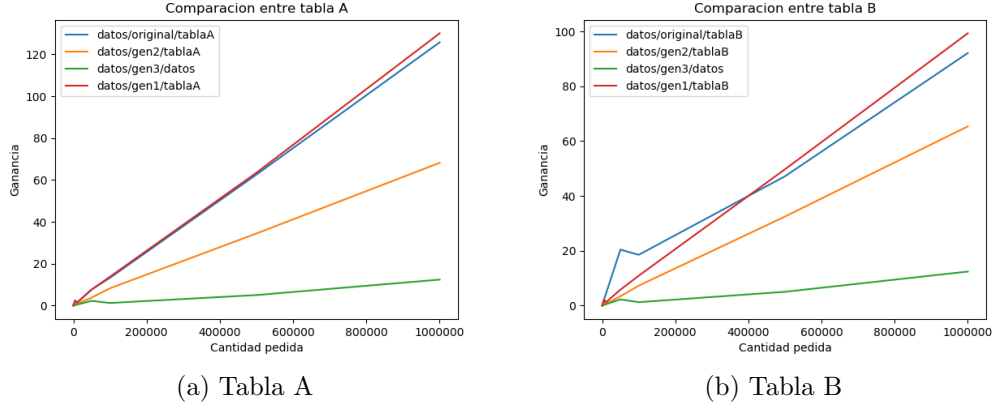


Figura 7: Diferencias de tiempo entre las distintas mejoras.

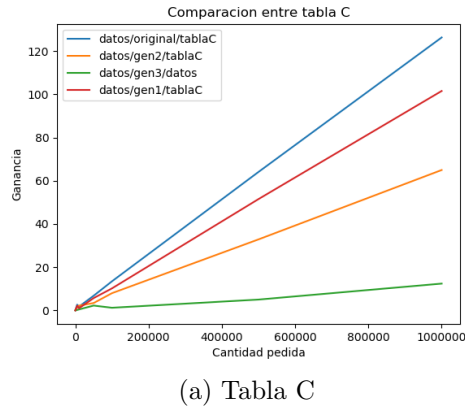


Figura 8: Diferencias de tiempo entre las distintas mejoras.

Al general la tabla con los datos de forma decreciente en la distribución triangular vamos a notar una gran mejora respecto a generarla de forma normal (Figura 8a). Aunque sigue siempre mas lento que si usáramos la búsqueda binaria que funciona de forma muy rápida.

Por lo tanto si tuviéramos que elegir algún método yo usaría la mejora de usar la búsqueda binaria. Por que nos permite tener distintas distribuciones pero sin gastar excesivo tiempo a la hora de generar los datos. La generación en tiempo constante aunque es muy rápida nos impide usar distintas distribuciones por lo tanto no seria una opción adecuada para segun que tiempo de modelo de Monte Carlo.

2.2. Generadores congruenciales

En esta sección veremos como funciona los generadores de datos aleatorios congruenciales. Para ello haremos pruebas con diferentes generadores ademas de usar dos a diferentes. Esto lo hacemos para ver como afecta las distintas variables a los generadores y como el efecto de la semilla.

- $a_1 = 2061$
- $a_2 = 2060$

En todos los generadores y pruebas fijaremos semilla inicial a $x = 19$, para que esto no influya a la hora de generar.

Tipo Generador	a	Semilla	Ciclos
Usando %	2061	19	1000
Usando %	2060	19	4
Usando aritmetica real	2061	19	78
Usando aritmetica real	2060	19	6
Usando aritmetica real corregida	2061	19	1000
Usando aritmetica real corregida	2060	19	4
Usando fmod	2061	19	1000
Usando fmod	2060	19	4

Tabla 6: Modificación 2

Como podemos ver en la Tabla 6 todos los generadores funcionan de forma parecida. Todos nos dan buenos resultados exceptuando los de aritmética real sin corregir. Para que un generador de este tipo y que funcione es necesario que c y m no tienen que tener divisores en común. Ademas $a-1$ tiene que ser múltiplo de todos los divisores primos de m .

3. Fe de erratas

En las Figuras 7a, 7b y 8a las leyendas de las graficas están mal. Donde pone Ganancia debería ser tiempo y en Cantidad pedida seria la cantidad de números pedidos al generador.