Simulación de Sistemas 2022-2023
Grado en Ingeniería Informática
Universidad de Granada

Memoria Práctica 2

Autor:
Brian Sena Simons.

Grupo:
3ºA Subgrupo A2

Brian,
Sena Simons

Modelo de Monte Carlo

Ingeniería Informática
8 de noviembre de 2022

Índice

1. Problema del Quiosco

1.1. Introducción

Un establecimiento (por ejemplo un quiosco de periódicos) se abastece diariamente de un cierto producto, y necesita decidir cuántas unidades de ese producto pedir cada día. Este valor, s, se fija en un contrato a largo plazo con el proveedor (y una vez fijado no puede cambiarse). El establecimiento obtiene una ganancia de x euros por cada unidad vendida, con un costo de y euros por cada unidad comprada. La demanda D (o sea, el número de unidades del producto que se solicitan cada día) varía diariamente, pero se ha estudiado que se ajusta a una determinada distribución de probabilidad P (D = d), que se detallará después. Se desea encontrar el valor óptimo de s, donde el criterio de optimalidad es maximizar la ganancia esperada.

1.2. Definición de funciones

En términos más formales, el problema se describirá sobre todo con las siguientes funciones:

$$g(x,y,s,d) = \begin{cases} x \cdot s - y \cdot s, & d \geq s \\ x \cdot d - y \cdot s, & d < s \end{cases} \tag{1.1}$$

Dónde tendremos diversas probabilidades de demanda:

$$\begin{aligned} P_a(D=d) &= \tfrac{1}{100}, \forall d = 0, ..., 99 \\ P_b(D=d) &= \tfrac{d+1}{5050}, \forall d = 0, ..., 99 \\ P_c(D=d) &= \begin{cases} \frac{d}{2500} & 0 \leq d < 50 \\ \frac{100-d}{2500} & 50 \leq d \leq 99 \end{cases} \end{aligned} \tag{1.2}$$

Y a partir de ellas podremos definir la ganacia esperada como la media de la función de ganancia diaria respecto de una distribución P(D=d), es decir:

$$G(x,y,s) = \sum_{d}^{99} g(x,y,s,d) \cdot P(D=d) \tag{1.3}$$

Y por tanto nuestro problema consiste en encontrar el valor óptimo para S, tal que nuestra ganancia es máxima.

1.3. Análisis Teórico

Si estudiamos[1] las fórmulas definidas anteriormente podemos derivar la función en concreto a maximizar por nuestro modelo y obtener el valor S de forma análitica. Ya que en concreto, para la probabilidad A tendríamos que maximizar la siguiente función:

$$G_a(x,y,s) = -\frac{200sy + (s^2 - 199s)\,x}{200} \tag{1.4}$$

Que es resultado de simplificar la sumatoria definida en el apartado anterior sustituyendo. Si hacemos esto para todas las probabilidades obtendremos las siguientes ecuaciones:

$$\begin{aligned}
G_a(x,y,s) &= -\frac{200sy+\left(s^2-199s\right)x}{200} \\
G_b(x,y,s) &= \frac{30300sy+\left(s^3+3s^2-30298s\right)x}{30300} \\
G_c(x,y,s) &= \begin{cases} -\frac{\left(6s^2+14994s\right)y+\left(s^3-6s^2-14995s\right)x}{15000} & 1 \le d < 50 \\ \frac{\left(6s^2-15906s\right)y+\left(s^3-306s^2+30605s-234900\right)x}{15000} & 50 \le d \le 99 \end{cases}
\end{aligned} \tag{1.5}$$

Para obtener el valor óptimo de la ganancia para un x é y fijos solamente necesitamos derivar respecto a s. Ahora si igualamos a cero cada una de las derivadas y despejamos s, obtenemos ecuaciones que determinan máximos y mínimos de las ecuaciones que se pueden comprobar

[1] Con este documento, se adjunta un fichero ".wxmx"de máxima en el cuál se desarrollan las fórmulas que aquí se obtienen

luego con la segunda derivada [2]. El resultado sería:

$$\begin{aligned}
\frac{\nabla G_a(x,y,s)}{\nabla s} &= -\frac{200y+(2s-199)x}{200} \\
\frac{\nabla G_b(x,y,s)}{\nabla s} &= -\frac{30300y+\left(3s^2+6s-30298\right)x}{30300} \\
\frac{\nabla G_c(x,y,s)}{\nabla s} &= \begin{cases} -\frac{(12s+14994)y+\left(3s^2-12s-14995\right)x}{15000} & 1 \leq d < 50 \\ \frac{(12s-15906)y+\left(3s^2-612s+30605\right)x}{15000} & 50 \leq d \leq 99 \end{cases}
\end{aligned} \tag{1.6}$$

Si hacemos los cálculos, obtendremos la tabla de máximos para un conjunto de valores x é y fijos que se enseña en el anexo.

1.4. Resultados de simulación

Ahora veremos que somos capaces de estimar el mismo valor de S que en el análisis teórico de forma mucho más sencilla al simular un entorno que cumpla con las condiciones de nuestro problema.

El primer paso es determinar el bucle principal de nuestro modelo, dónde en ese bucle, que se repetirá N veces, realizaremos una llamada a nuestro generador de datos correspondiente a la distribución de probabilidad que estamos estudiando, y luego realizaremos el cálculo de la ganancia diaria obtenida en esa iteración, esa ganancia se irá acumulando N veces. Una vez terminada las N iteraciones, que representan días, obtendremos la ganancia media. Esa es nuestra ganancia esperada, que tendrá una desviación según el número de iteraciones que habremos hecho.

Ese bucle que repetiremos N veces, lo haremos para valores de $s \in [0, 99]$ para buscar el mejor valor de "s" dentro de nuestro problema. Los resultados obtenidos son:

[2] Se deja al lector que verifique en máxima o que realize los cálculos

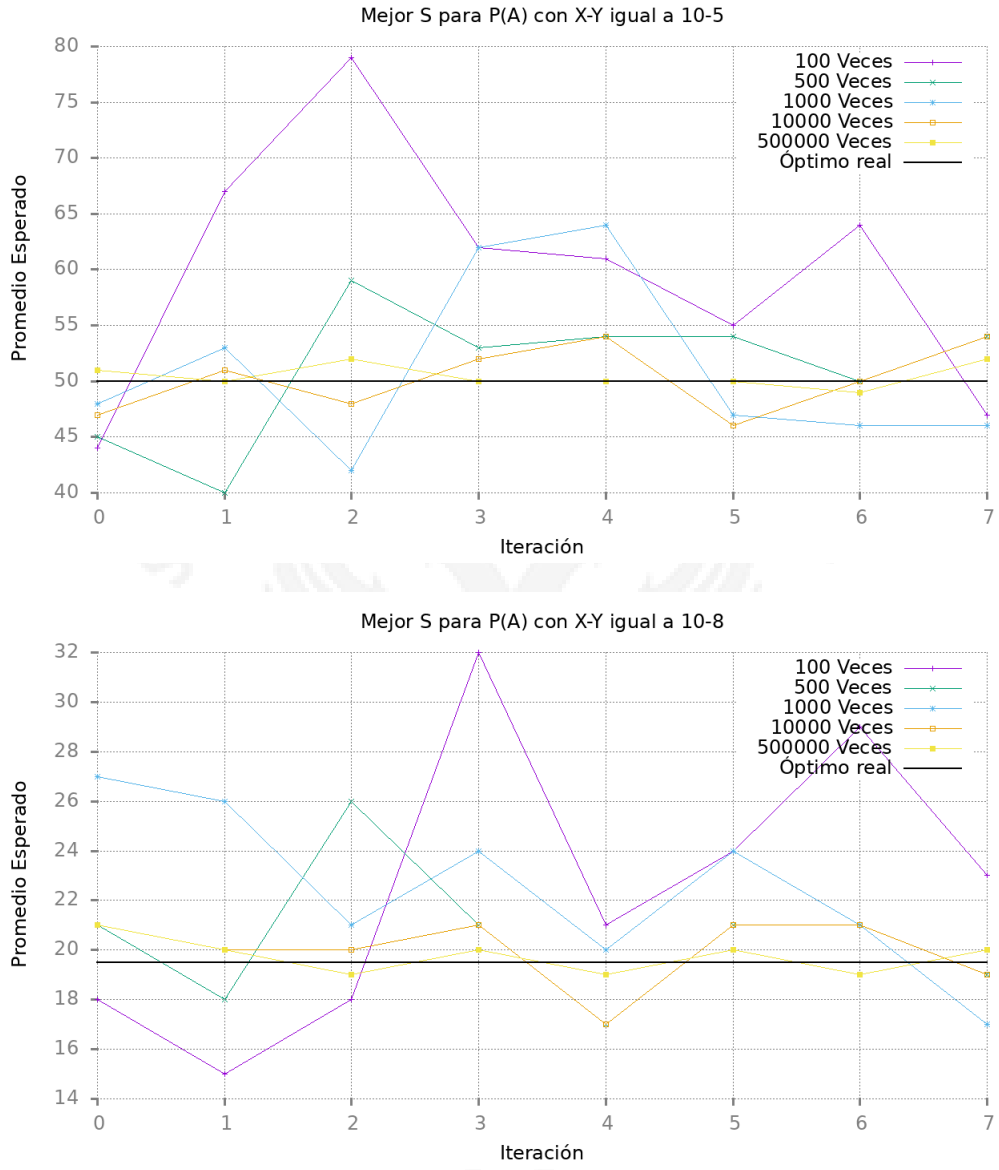


Figura 1.1: Como vemos, el número de veces en el cual repetimos el bucle principal conlleva a resultados más cercanos al óptimo real (línea en negrita).

La visualización para las distintas políticas y valores es análoga, cuanto mayor el número de repeticiones más cercano al valor real nos encontramos. Como vemos, hemos podido obtener una aproximación, utilizando un valor de N no tan grande, bastante cercano al valor real. (Ver tablas en el anexo)

La parte más complicada a la hora de implementar un modelo de simulación de Monte Carlo es la necesidad de tener un buen generador de datos que se acerca a la probabilidad real de nuestro problema. Entre otras palabras, esto nos enseña la posibilidad de aproximar problemas en los cuales resolver análiticamente no resulta ser tan sencillo como en este caso. Es justo lo que se aprecia en la siguiente sección.

1.5. Política de devolución

Si ahora implementamos la posibilidad de devolver las unidades compradas no vendidas en nuestro modelo, donde podemos devolver, o no, nuestras unidades no vendidas pagando un precio fijo z, obtenemos una nueva función de ganancia diaria que sigue:

$$g(x,y,s,z,d) = \begin{cases} x \cdot s - y \cdot s & d \geq s \\ x \cdot d - y \cdot d - min\{z, y \cdot (s-d)\} & d < s \end{cases} \tag{1.7}$$

Como vemos, empezamos a subir la complejidad del problema a resolver de forma analítica [3]. Ahora nuestra función de ganancia esperada tendría una forma más compleja:

$$G(x,y,s,z) = \begin{cases} \frac{\left(s^2-100s\right)y+\left(100s-s^2\right)x}{100} & d > s \\ d < s & \begin{cases} -\frac{2sz+\left(s^2-s\right)y+\left(s-s^2\right)x}{200} & z < y \cdot (s-d) \\ -\frac{2s^2y+\left(s-s^2\right)x}{200} & y \cdot (s-d) < z \end{cases} \end{cases} \tag{1.8}$$

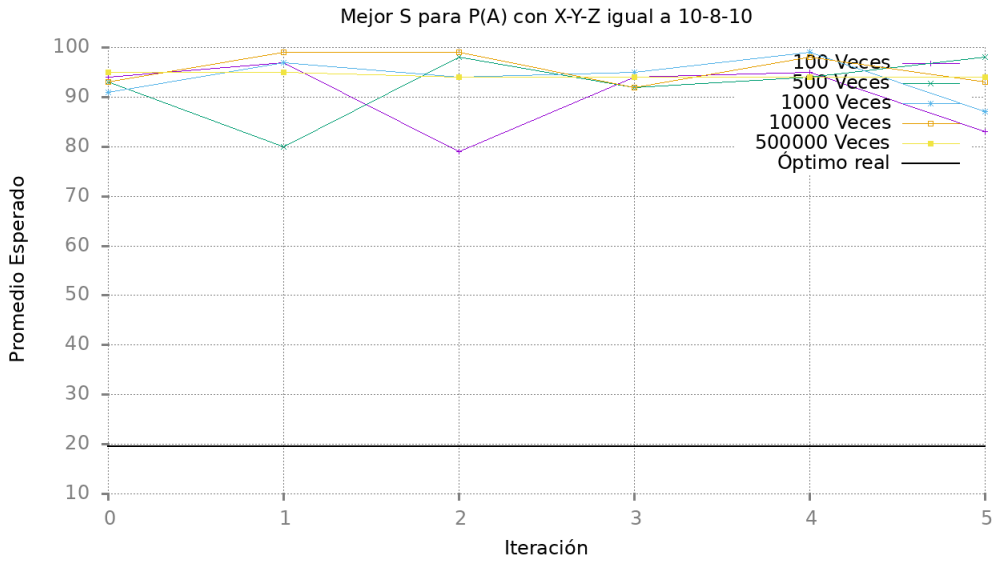
Ya no es tan sencillo encontrar el máximo de esa función, y eso que estamos hablando de la distribución $P_a(D=d)$ que es, a igual que $P_b(D=d)$, de las más sencillas. Ya que el hecho de meter la $P_c(D=d)$ implicaría dividir aún más nuestras sumatorias. Para problemas como este es cuando podría convenir emplear directamente un modelo de simulacion Monte Carlo.

[3] En nuestro archivo de máxima también se aprecia la dificultad de plasmar el nuevo sumatorio resultante de esta función

1.6. Simular con devoluciones

Si realizamos la simulación con devoluciones para los mismos X e Y fijados anteriormente, ahora solo tendríamos que jugar con el valor de Z para observar el comportamiento. Es trivial ver, por la definición de la función, que si nuestro valor Z es muy bajo, obtendremos que la mejor política será siempre comprar el máximo número de periódicos ya que el coste de perder un cliente podría ser mayor que el de devolver las unidades no vendidas. Mientras que para valores muy grandes deberíamos obtener un comportamiento similar al de las funciones anteriores.

Resulta ser que es justo ese comportamiento el que obtenemos si utilizamos los siguientes valores $z \in [10, 100, 1000]$. Para el primero, dado que es indiferente al número de unidades sin vender, el comportamiento es que compremos todas las unidades. Para el segundo empezamos a ver que baja el número de unidades a comprar. Y para el último obtenemos exactamente el mismo comportamiento que en la política anterior sin devouciones.



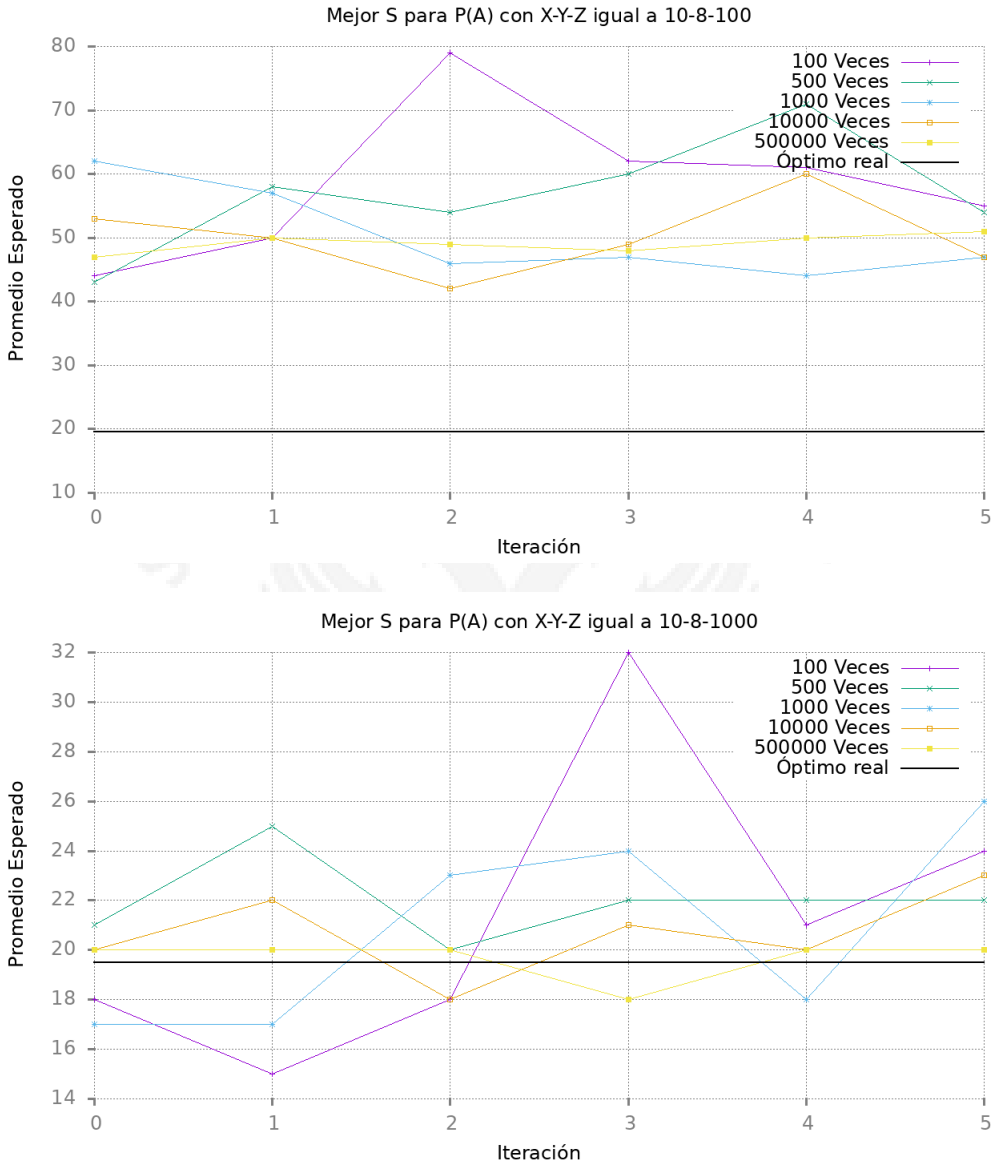


Figura 1.2: El valor de la devolución implica cambios en el comportamiento como los descritos anteriormente

¡Aunque, no es cierto que se cumpla con los mismos valores para las distintas probabilidades! Para alguno de las combinaciones de valores X,Y é P_b ó P_c, el comportamiento es similar pero no idéntico.

Pero sí que se observa la facilidad en la cual hemos podido analizar el problema y obtener un

máximo aproximado con cierta fiabilidad. Dado que el único cambio que tuvimos que hacer al código anterior ha sido modificar la función para cuando $d < s$ e introducir una nueva variable z.

2. Optimizar generadores de datos

Una parte del tiempo que ha conllevado realizar la simulación ha sido debido a la generación de los datos acorde la función de distribución en cada apartado. Es por ello que en este apartado se incita a estudiar e intentar mejorar el rendimiento de los generadores.

La primera mejora propuesta es ordenar la tabla en orden decreciente antes de construirla. La segunda mejorar propuesta es utilizar el famoso método de búsqueda binaria y por último, para el apartado $P_a(D = d)$ podemos hacer la búsqueda constante.

Los resultados de la primera mejora se puede apreciar con el siguiente gráfico de rendimiento.

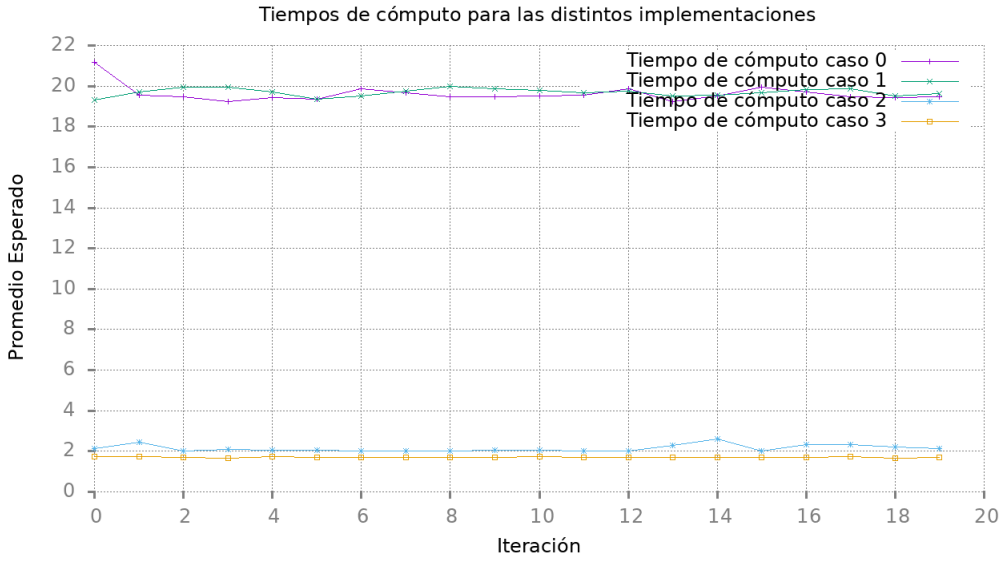


Figura 2.1: Diferencias en tiempo de cómputo, en millisegundos, para las distintas implementaciones caso A.

Vemos que claramente podemos reducir la probabilidad uniforme del caso A, a una constante de la forma $u/(1/n)$. Y Luego en caso B, que también es representativo del caso C, se ve también una clara reducción del tiempo de cómputo si reordenamos las probabilidades (Reducción casi a la mitad) y luego una mejora sustancial si utilizamos un método de búsqueda avanzado, como es la búsqueda binaria.

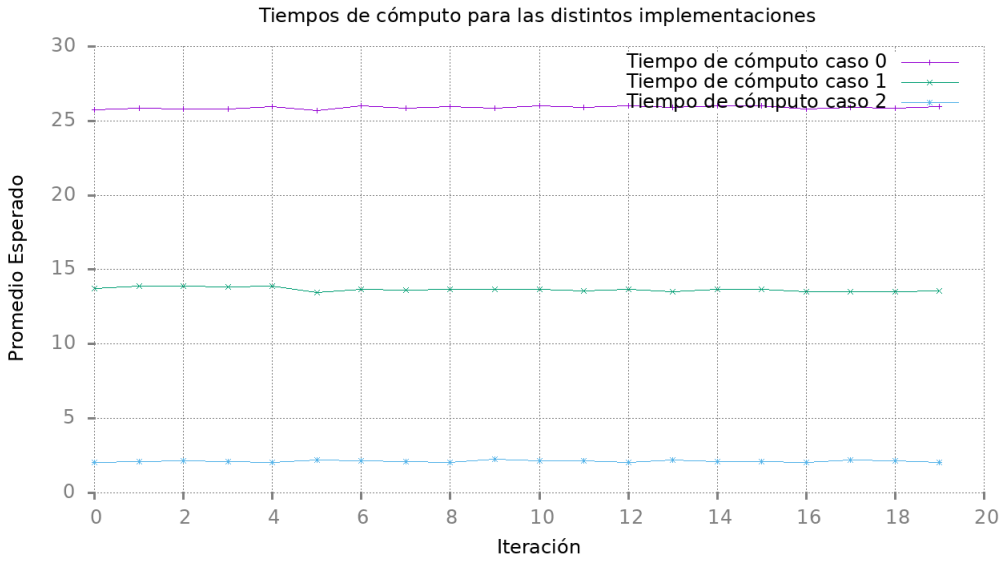


Figura 2.2: Diferencias en tiempo de cómputo, en millisegundos, para las distintas implementaciones caso B.

A todo esto, conviene comprobar que las distribuciones de probabilidades no se ven afectadas cuando realizamos estos cambios, tanto en la búsqueda como en el orden los segmentos. La conservación de la distribución se puede apreciar en las gráficas que están en el anexo.

3. Generadores básicos

En esta sección vamos a implementar un generador congruencial lineal multiplicativo y estudiar sus características. El generador en concreto es:

$$x_{n+1} = (1013 \cdot x_n) \, mod \, m \tag{3.1}$$

Con $m = 2^{12}$. Dónde utilizaremos un algunas variaciones como:

Algorithm 1 aritmética entera

$x = (1013 \cdot x) \, \%m$

Algorithm 2 Aritmética real

$x = (1013 \cdot x) \, \%m$
$x = (x - (int)x) \cdot m$

Algorithm 3 Airtmética real "artesanal"

$1013 \cdot x/m$
$(x - (int)x) \cdot m$

Algorithm 4 Aritmética real usando "fmod"

$x = fmod(1013 \cdot x, m)$

Donde el objetivo es obtener el máximo periodo para la sucesión de números aleatorios, ya que siempre conviene utilizar una fuente de sucesión grande, donde se obtengan muchos más números que los necesarios para la aplicación específica. Pero por definición, y por el teorema de Hull-Dobell, sabemos que ninguna de las variaciones debería llegar a alcanzar el máximo. En este caso el máximo es $M = 2^{12}$.

La razón por la cuál ninguno debería llegar al máximo es porque incumplen con el teorema que impone que para ser máximal, c y m deben ser primos relativos. En nuestro caso $m =$

$2^{12}, c = 0$, por lo que ya nos indica que no llegaremos a alcanzar el máximo. Pero si es cierto, que podemos obtener un periodo submáximo determinado analíticamente dado que $M = 2^e$ es una potencia de dos y a, el factor que multiplica a la entrada, es de la forma $a = 8 \cdot K - 5 = 1013$. Llegaremos entonces a un submáximo $2^{e-2} = M/4$.

Hay que tener en cuenta un par de cosas, al ser un modelo congruencial lineal multiplicativo, debemos forzar $x_0 \neq 0$. Al usar flotantes y/o dobles en algunos modelos, obtendremos periodos mayores de los reales según el nivel de granularidad en el cual bajemos. Si usamos un filtro, δ, que sea la mínima diferencia para que dos valores flotantes se consideren iguales, ese filtro puede afectar los periodos esperados. Ya que una mayor precision conlleva a mayores periodos.

Si establecemos un intervalo de precisión de 4 decimales, obtenemos las siguientes tablas:

Periodos obtenidos				
Entero	Real Flotante	Real Doble	Corregida	Fmod
721.28	2221.94	5729.57	711.10	5477.09
700.96	2238.09	5335.50	699.92	5754.97
84.80	2266.11	5542.35	706.72	5410.44
659.68	2599.44	6079.05	746.56	5612.10
672.06	2385.90	5851.41	698.56	5908.81

Tabla 3.1: Periodos promedios para 100 medidas, con un máximo de 500000, realizadas 5 veces con semillas diferentes

Y como vemos, los números enteros se acercan bastante al máximo teórico. La razón por la cuál no llegaban al máximo es por que no cumpliamos con una de las restricciones, y era de que x_0 fuera impar. Si lo forzamos tenemos:

Periodos obtenidos				
Entera	Real flotante	Real doble	Corregida	Fmod
1024.00	2361.01	5820.05	692.56	6386.91
1024.00	2343.56	5416.34	705.60	5537.71
1024.00	2485.71	5890.99	711.52	5548.50
1024.00	2483.68	5954.13	672.96	6051.22
1024.00	2558.46	5937.46	671.60	5645.91

Tabla 3.2: Resultados de inicializar los valores a números impares

Además, si miras la secuencia de 10 números cercanos a dónde ocurrió la repetición vemos que los números aleatorios no parecen triviales de adivinar:

Vecindario de números generados						
Entera	3039	2411	1127	2963	3247	123
Real flotante	3532.23	2345	3901	3169	3029	473
Real doble	3918.26	174.513	653.932	2977.5	1550.92	2310.47
Corregida	3639	4003	4095	3083	1927	2355
Fmod	3256.5	1555.58	2940.73	1166.93	2454.58	219.418

Dado el buen comportamiento aleatorio de las diversas implementaciones, a la hora de elegir el generador de números aleatorios deberíamos elegir aquel que maximize el periodo. En nuestro caso tanto el fmod como la aritmética real que utiliza dobles nos serviría.

4. Anexo

4.1. Óptimos reales sin devolución

X = 10, Y = 1	89.5
X = 10, Y = 5	50
X = 10, Y = 8	19.5

Tabla 4.1: Resultados para la demanda generada por $P_a(D = d)$

X = 10, Y = 1	94.34
X = 10, Y = 5	70.06
X = 10, Y = 8	43.94

Tabla 4.2: Resultados para la demanda generada por $P_b(D = d)$

X = 10, Y = 1	75
X = 10, Y = 5	51
X = 10, Y = 8	32

Tabla 4.3: Resultados para la demanda generada por $P_c(D = d)$

4.2. Optimos Obtenidos con Monte Carlo

X = 10, Y = 1	89.5
X = 10, Y = 5	50.5
X = 10, Y = 8	19.75

Tabla 4.4: Resultados para la demanda generada por $P_a(D = d)$

X = 10, Y = 1	94.375
X = 10, Y = 5	69.875
X = 10, Y = 8	43.75

Tabla 4.5: Resultados para la demanda generada por $P_b(D = d)$

X = 10, Y = 1	78.5
X = 10, Y = 5	50.125
X = 10, Y = 8	31.75

Tabla 4.6: Resultados para la demanda generada por $P_c(D = d)$

4.3. Función de distribución conservada

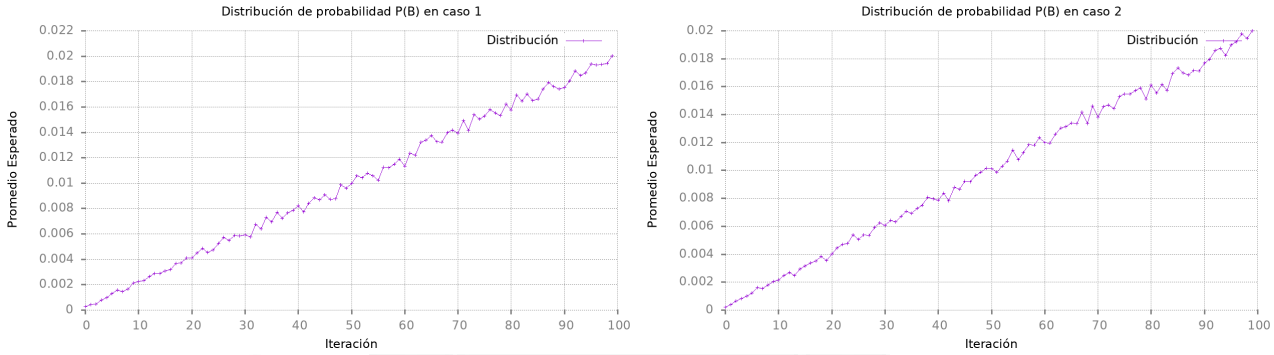


Figura 4.1: Preservación de la función de distribución al re-ordenar y al utilizar búsqueda binaria en el caso B.

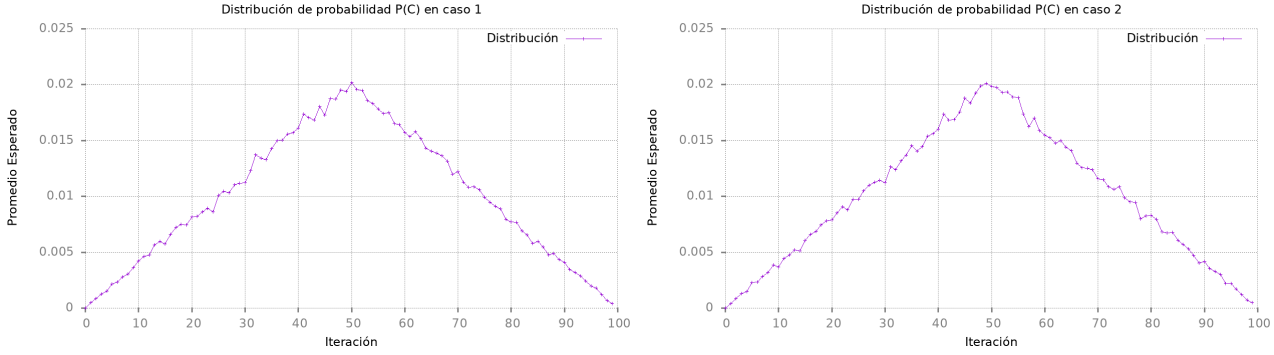


Figura 4.2: Preservación de la función de distribución al re-ordenar y al utilizar búsqueda binaria en el caso C.

Referencias