Lista 3 – Modelagem e Simulação

Grupo: Lucas Moreira Magalhães
Marcus Adriano Pereira
Vitor Hugo Honorato Tiago

Questão 3.1.1 -
(a) Modify program ssq2 to use Exponential (1.5) service times.
(b) Process a relatively large number of jobs, say 100 000, and determine what changes this produces relative to the statistics in Example 3.1.3?
(c) Explain (or conjecture) why some statistics change and others do not.

Resposta -
(a) Código C em anexo.

(b) Comparando as estatísticas temos que a média do tempo de chegada, média do tempo de serviço e a utilização do servidor são iguais nos dois exemplos, as demais estatísticas possuem valores maiores quando utilizamos a função exponencial. Abaixo temos as estatísticas.

	Uniforme	Exponencial
r_barra	2.0	2.0
w_barra	3.85	6.04
d_barra	2.35	4.53
s_barra	1.5	1.5
l_barra	1.92	3.02
q_barra	1.17	2.27
x_barra	0.75	0.75

(c) O r_barra não foi modificado pois não houve alteração na geração dos tempos de chegada. O s_barra não foi modificado pois a função exponencial utilizada tem argumento 1.5, então quando temos um número de Jobs grande a média do tempo de serviço gerado tende a se igualar ao argumento, portanto 1.5. A utilização é dependente do r_barra e s_barra, logo se esses não sofrem alteração, x_barra também não sofrerá alteração. As demais estatísticas aumentam quando é utilizada a função exponencial, isso devido a maior probabilidade da exponencial de gerar números maiores. Apesar da média convergir pra 1.5 em ambas as funções, na função exponencial temos uma maior chance de gerar números grandes, o que aumenta as estatísticas de delay e wait, que influenciam no calculo de l_barra e q_barra.

Questão 3.1.3 -
(a) Given that the Lehmer random number generator used in the library rng has a modulus of 2^31−1, what are the largest and smallest possible numerical values (as a function of μ) that the function Exponential(μ) can return?

(b) Comment on this relative to the theoretical expectation that an Exponential (μ) random variate can have an arbitrarily large value and a value arbitrarily close to zero.

Resposta -
(a) Seguindo as fórmulas de geração de número aleatório do rng e da distribuição exponencial temos que x, sendo exponencial é definido por:

$$x = -\mu \ln(1 - random)$$

Sendo random o número aleatório gerado em rng. Esse número aleatório varia entre 0 < random < 1, logo temos que seus limites são inferior = $\frac{1}{2^{31}-1}$ e superior = $\frac{2^{31}-2}{2^{31}-1}$. Portanto, utilizando o limite superior temos:

$$x = -\mu \ln(1 - \frac{2^{31}-2}{2^{31}-1})$$

E pelo limite inferior temos:

$$x = -\mu \ln(1 - \frac{1}{2^{31}-1})$$

(b) Podemos considerar que quanto maior o módulo m utilizado, $\frac{m-1}{m}$ tende a ser 1, e $\frac{1}{m}$ tende a ser 0. Portanto, os limites da função random serão 0 e 1. Utilizando a fórmula da geração exponencial, temos que:

$$x = -\mu \ln(1 - 1) \quad \text{e} \quad x = -\mu \ln(1 - 0)$$

Considerando que ln(0) tende a -∞ e ln(1) tende a 0, concluímos que os limites de x serão 0, quando utilizado o limite inferior de random, e ∞, quando utilizado o limite superior de random.

Questão 3.1.5 -
(a) Verify that the mean service time in Example 3.1.4 is 1.5.
(b) Verify that the steady-state statistics in Example 3.1.4 seem to be correct.
(c) Note that the arrival rate, service rate, and utilization are the same as those in Example 3.1.3, yet all the other statistics are larger than those in Example 3.1.3. Explain (or conjecture) why this is so. Be specific.

Respostas -
(a) Código C em anexo.

(b) Nossos resultados foram iguais nas médias do tempo de serviço, tempo de chegada e na utilização. Nas demais estatísticas houve um pequeno aumento, devido a diferenças dos números gerados nas duas simulações.

	Exemplo	Simulação
r_barra	2.0	2.0
w_barra	5.77	6.02
d_barra	4.27	4.53
s_barra	1.5	1.49
l_barra	2.89	3.02
q_barra	2.14	2.27
x_barra	0.75	0.75

(c) Na nova simulação feita é utilizada uma distribuição geométrica que depende de um argumento p, o que não acontecia no exemplo 3.1.3. A média dos números gerados por Geometric(p) tende a ser $\frac{p}{1-p}$, como na simulação é utilizado p =0.9, temos que a média dos números gerados tende a ser alta. Portanto, temos um maior delay e wait, consequentemente um maior l_barra e q_barra.

Questão 3.1.6 -

(a) Modify program sis2 to compute data like that in Example 3.1.7. Use the functions PutSeed and GetSeed from the library rng in such a way that one initial seed is supplied by the system clock, printed as part of the program's output and used automatically to generate the same demand sequence for all values of s.

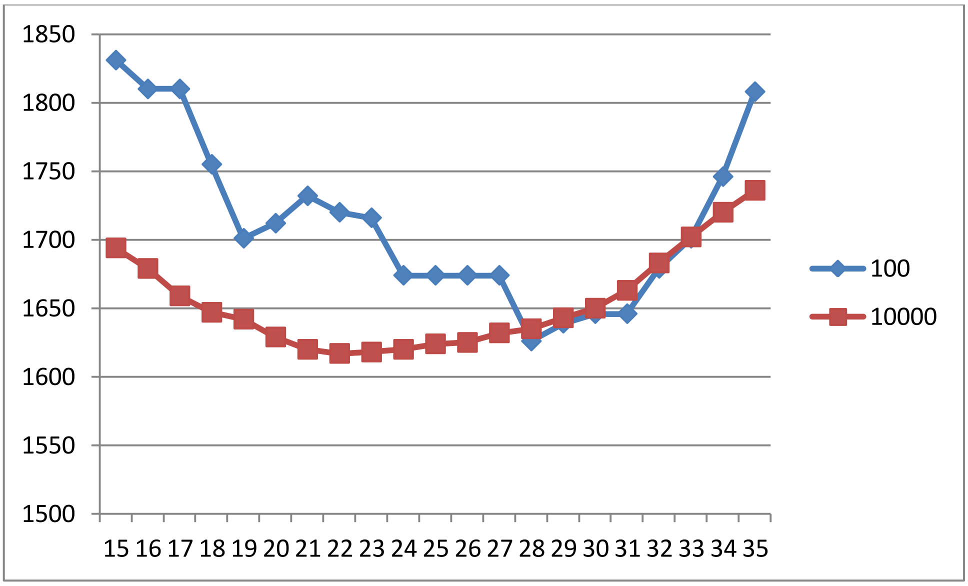
(b) For s = 15, 16, . . . , 35 create a figure (or table) similar to the one in Example 3.1.7.

(c) Comment.

Respostas -

(a) Código C em anexo.

(b)



(c) Visto que no início do código é gerada uma semente aleatória, mas essa é utilizada para todos os valores de s, podemos comparar esses valores entre si. Como visto no gráfico na letra (b), quando n=100 temos dados um pouco menos organizados e o mínimo custo é alcançado com s=28, entretanto quando n=10000 temos uma parábola evidente no gráfico e o mínimo custo é alcançado com s=22. Como houve mais tempo para simulação em n=10000, é mais confiável levar em consideração essa simulação onde o mínimo está em s=22.

Questão 3.1.7 -

(a) Relative to Example 3.1.5, if instead the random variate sequence of demands are generated as d i = Equilikely(5, 25) + Equilikely(5, 25) i = 1, 2, 3, . . . then, when compared with those in Example 3.1.6, demonstrate that some of the steady-state statistics will be the same and others will not.

(b) Explain why this is so.

Resposta –

(a) Código C em anexo. Os dados obtidos foram:

	Exemplo	Simulação
d_barra	30.00	29.56
o_barra	30.00	29.56
u_barra	0.39	0.39
l+_barra	42.86	43.53
l-_barra	0.26	0.17

(b) Pode-se notar que as estatísticas d_barra, o_barra e u_barra permaneceram as mesmas. Isso devido ao cálculo da média da distribuição Equilikely ser (a+b)/2, assim (10+50)/2 = (5+25)/2 + (5+25)/2. Houve uma pequena mudança nas estatísticas l+_barra e l-_barra. Isso acontece devido a maior discrepância de valores quando utilizado a soma dos Equilikely, pois apesar da média ser a mesma, os valores tem um maior desvio padrão.

Questão 3.2.3 -

Modify program ssq2 as suggested in Example 3.2.7 to create two programs that differ only in the function GetService. For one of these programs, use the function as implemented in Example 3.2.7; for the other program, use

double GetService(void) { SelectStream(2); /* this line is new */ return (Uniform(0.0, 1.5) + Uniform(0.0, 1.5)); }

(a) For both programs verify that exactly the same average interarrival time is produced (print the average with d.dddddd precision). Note that the average service time is approximately the same in both cases, as is the utilization, yet the service nodes statistics w , d, l, and are different.

(b) Why?

Respostas –

(a)Código C em anexo. Os dados obtidos foram:

	Exemplo	Simulação
r_barra	1.995367	1.995367
w_barra	3.86	4.07
d_barra	2.36	2.58
s_barra	1.50	1.49
l_barra	1.93	2.04
q_barra	1.18	1.29
x_barra	0.75	0.75

(b) O tempo médio de serviço, chegada e utilização são os mesmos devido a maneira de calcular a média da distribuição Uniforme, ou seja:

$$\frac{(1+2)}{2} = \frac{(0+1.5)}{2} + \frac{(0+1.5)}{2} = 1.5$$

As outras estatísticas se diferem devido ao maior desvio padrão quando é utilizada a soma de duas distribuições Uniforme.