Universidade Federal de Uberlândia

Modelagem e Simulação

Equipe: Bruno Borges
Jefferson Oliveira
Marilia Leal
Vinicius Scavoni

Uberlândia – 2017

3.1.1.
(a) Modify program ssq2 to use Exponential(1.5) service times.

Mudança no código.

(b) Process a relatively large number of jobs, say 100000, and determine what changes this produces relative to the statistics in Example 3.1.3?

A diferença está no tempo médio de espera, tempo médio de delay, média de jobs no nó de serviço e média de jobs na fila.

Resultados exemplo 3.1.3

$\bar{r}$	$\bar{w}$	$\bar{d}$	$\bar{s}$	$\bar{l}$	$\bar{q}$	$\bar{x}$
2.00	3.83	2.33	1.50	1.92	1.17	0.75

Resultados exercício 3.1.1

```
for 100000 jobs
   average interarrival time =   2.00
   average wait ............ =   6.04
   average delay ........... =   4.53
   average service time .... =   1.50
   average # in the node ... =   3.02
   average # in the queue .. =   2.27
   utilization ............. =   0.75
```

(c) Explain (or conjecture) why some statistics change and others do not.

A média de tempo entre chegadas não mudou, pois essa variável depende apenas de GetArrival() que foi um trecho do código não modificado.
O tempo médio de serviço não mudou, pois quando tirada a média dos tempos de serviço gerados pela função Uniforme(a, b) utilizada no Exemplo 3.1.3, tal média converge para (a+b)/2 = 3/2 = 1.5, e no caso da média dos tempos de serviço gerados pela função Exponencial(u) utilizada no exercício 3.1.1, tal média converge para u, que no caso foi 1.5.
A taxa de utilização não mudou, pois depende da média dos tempos de serviço e do tempo de departure que também não teve modificações.
No resto houve modificações, pois dependiam dos valores de service que não foram os mesmos para ambas as funções.

3.1.2 – (a) Relative to the steady-state statistics in Example 3.1.3 and the statistical equations in Section 1.2, list all of the consistency checks that should be applicable. (b) Verify that all of these consistency checks are valid.

a) Precisamos verificar se os valores dado no exemplo faz sentido matematicamente com as formulas mostradas na seção 1.2.

1 – r = 2.00 - w = 3.83 - d = 2.33 - s = 1.50 – l = 1.92 - q = 1.17 - x = 0.75

- A media de jobs em fila como a soma da media de jobs em delay e media de jobs em serviço w = d+s
- O tempo médio no nó de serviço deve ser verificado e igual q + x

b)
Sabemos que o tempo médio de espera é: w = d + s, logo no steady-state o w deve apresentar essa característica:
3.83 = 2.33 + 1.50 – Valido

O tempo médio no nodo é dito a soma do tempo médio na fila com o tempo médio em serviço:
1.92 = 1.17 + 0.75 – Valido
3.1.3 -) A função exponencial(u) é o resultado de uma transformação não linear que pega os valores entre 0 e 1 e os transforma em valores entre 0 e infinito, e assim "esticando" o intervalo de grandes valores. Através deste método a probabilidade de geração de valores pequenos e grandes são iguais.

Exercise 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.

O menor valor é proximo a zero, mas nunca o próprio 0, o log em C não suporta tal resultado, e o máximo valor tende teoricamente ao infinito, sendo em cada caso próximo ao próprio μ.

Exercise 3.1.4 (a) Conduct a transition-to-steady-state study like that in Example 3.1.3 except for a service time model that is Uniform(1.3, 2.3). Be specific about the number of jobs that seem to be required to produce steady-state statistics. **(b)** Comment.

Seed→ 12345

```
for 3372180 jobs
   average interarrival time =   2.00
   average wait ............. =  10.11
   average delay ............ =   8.31
   average service time .... =   1.80
   average # in the node ... =   5.06
   average # in the queue .. =   4.16
   utilization .............. =   0.90

Process returned 0 (0x0)   execution time : 0.637 s
Press ENTER to continue.
```

Seed → 54321

```
for 2154200 jobs
   average interarrival time =   2.00
   average wait ............. =  10.13
   average delay ............ =   8.33
   average service time .... =   1.80
   average # in the node ... =   5.06
   average # in the queue .. =   4.16
   utilization .............. =   0.90

Process returned 0 (0x0)   execution time : 0.429 s
Press ENTER to continue.
```

Seed → 2121212

```
for 2745060 jobs
   average interarrival time =   2.00
   average wait ............. =  10.18
   average delay ............ =   8.38
   average service time .... =   1.80
   average # in the node ... =   5.09
   average # in the queue .. =   4.19
   utilization .............. =   0.90

Process returned 0 (0x0)   execution time : 0.610 s
Press ENTER to continue.
```

(b) Vemos que a convergência pro **steady-state** é lenta e varia bastante em algumas casas e que algumas sementes gera valores aleatórios que convergem mais rapidamente para o estado estacionário.

Exercise 3.1.5 (a) Verify that the mean service time in Example 3.1.4 is 1.5.

```
average service time .... =   1.50
```

A média no exemplo 3.1.4 é 1.5.

(b) Verify that the steady-state statistics in Example 3.1.4 seem to be correct.

```
for 500000 jobs
   average interarrival time =   2.00
   average wait ............ =   5.77
   average delay ........... =   4.27
   average service time .... =   1.50
   average # in the node ... =   2.89
   average # in the queue .. =   2.14
   utilization ............. =   0.75
```

Para 50000 jobs as estatísticas do exemplo 3.1.4 são iguais.

(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.

Estatísticas exemplo 3.1.3:

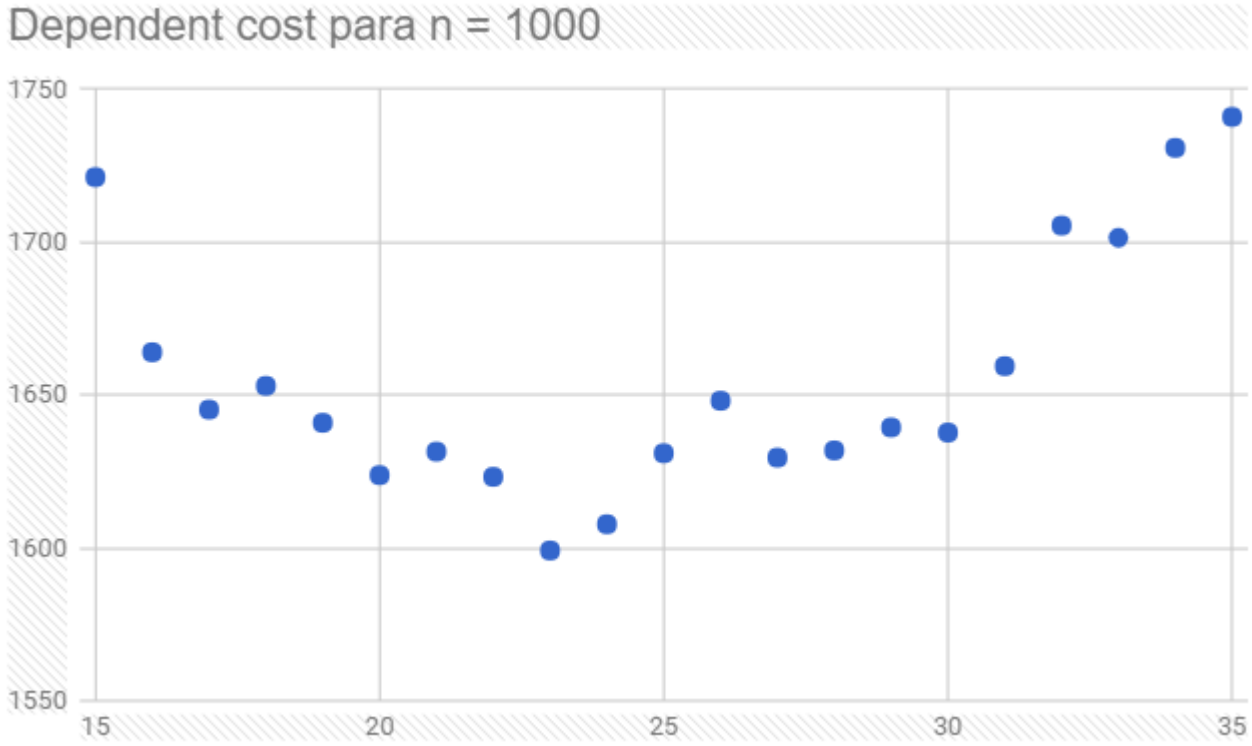
$\bar{r}$	$\bar{w}$	$\bar{d}$	$\bar{s}$	$\bar{l}$	$\bar{q}$	$\bar{x}$
2.00	3.83	2.33	1.50	1.92	1.17	0.75

As outras taxas no exemplo 3.1.4 serem maiores do que no exemplo 3.1.3 é dada pela sensibilidade das medidas de desempenho à distribuição do tempo de serviço. O que destaca a importância de usar um modelo de tempo de serviço preciso.

Exercise 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.

Verificar o código.

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



3.1.7 (a) Relative to Example 3.1.5, if instead the random variate sequence of demands are generated as di = 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 steadystate statistics will be the same and others will not.

Seed → 12345

```
for 9170 time intervals with an average demand of  30.00
and policy parameters (s, S) = (20, 80)

   average order ............. =  30.00
   setup frequency ........... =   0.39
   average holding level .... =  42.53
   average shortage level ... =   0.19
   average level ............. =  42.34

Process returned 0 (0x0)   execution time : 0.004 s
Press ENTER to continue.
```

Seed → 54321

```
for 15530 time intervals with an average demand of  30.00
and policy parameters (s, S) = (20, 80)

   average order ..... =  30.00
   setup frequency ..... =   0.39
   average holding level .... =  42.64
   average shortage level ... =   0.18
   average level ..... =  42.46

Process returned 0 (0x0)   execution time : 0.006 s
Press ENTER to continue.
```

Seed → 2121212

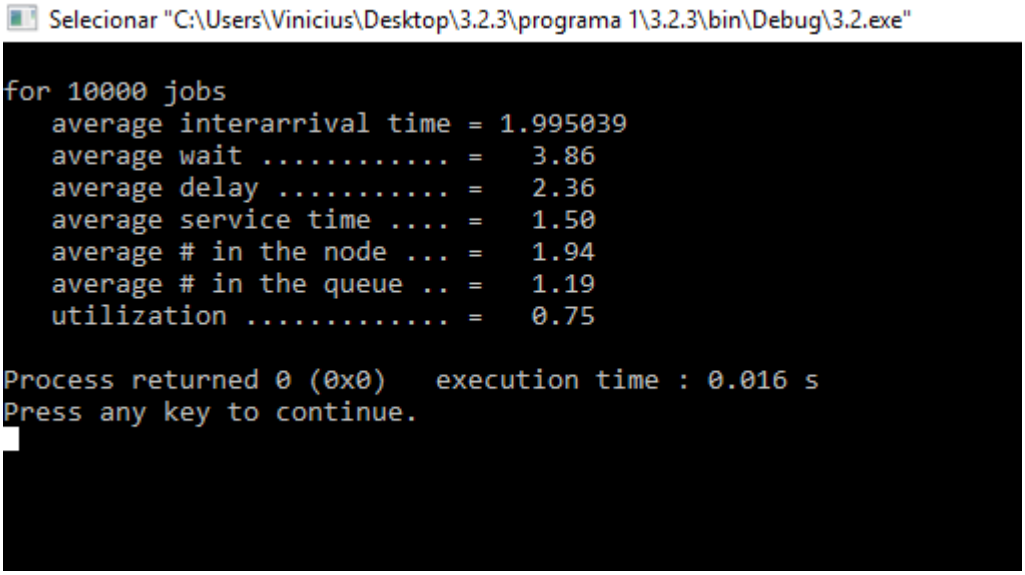
```
for 6980 time intervals with an average demand of  30.00
and policy parameters (s, S) = (20, 80)

   average order ..... =  30.00
   setup frequency ..... =   0.39
   average holding level .... =  42.58
   average shortage level ... =   0.18
   average level ..... =  42.39
```

(b) Explain why this is so

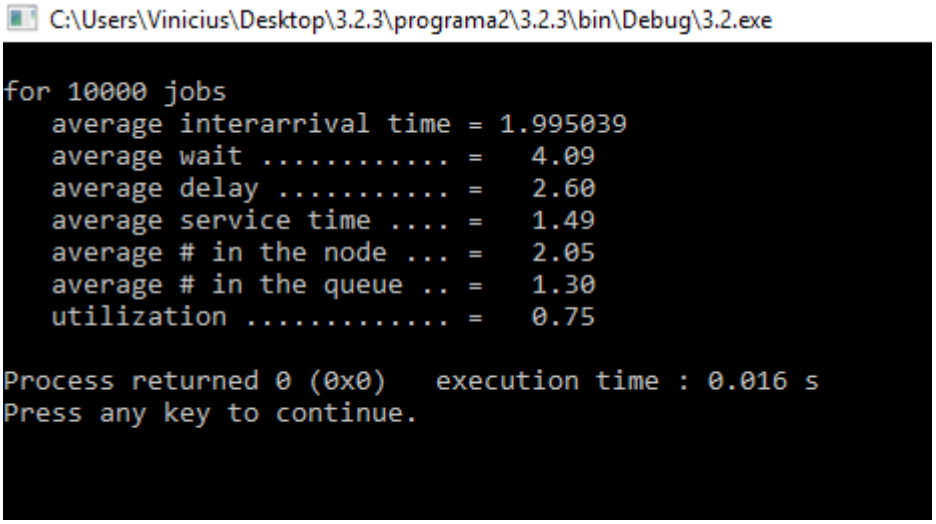
A diferença que vemos é um leve declínio no l- e l+ devido ao fato que os valores gerados pelos equilikelys do exercício tem um um intervalo menor que o intervalo gerado no exemplo devido a isso a funcao getDemand retorna valores menores do que os do exemplos.

Exercise 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 $\bar{w}$, d, $\bar{l}$, and $\bar{q}$ are different. (b) Why?

Programa 1



Programa 2



A e B-)O valores são parecidos pelas propriedades do método Uniform, nos dois o 0 é quase ignorado, na implementação uniform (0, 1.5) + uniform (0, 1.5) os valores ficam mais próximos de 1,5 que é quase a média da implementação uniform (0, 2). Por isso a proximidade de resultados.

3.2.4. Modify program ssq2 as suggested in Examples 3.2.8 and 3.2.9.
Código 3.2.4

(a) What proportion of processed jobs are type 0?
5997

(b) What are $\bar{w}$, $\bar{d}$, $\bar{s}$, $\bar{l}$, $\bar{q}$, and $\bar{x}$ for each job type?

```
for 10000 jobs [0]
   average wait ............ =   7.76
   average delay ........... =   5.76
   average service time .... =   1.99
   average # in the node ... =   1.95
   average # in the queue .. =   1.45
   utilization ............. =   0.50

for 10000 jobs [1]
   average wait ............ =   7.71
   average delay ........... =   5.73
   average service time .... =   1.99
   average # in the node ... =   1.30
   average # in the queue .. =   0.96
   utilization ............. =   0.33
```

(c) What did you do to convince yourself that your results are valid?

Checagem de consistência:

- Para os jobs do tipo 0:

  $\bar{w} = \bar{d} + \bar{s} = 5.76 + 1.99 = 7.75$
  $\bar{l} = \bar{q} + \bar{x} = 1.45 + 0.50 = 1.95$

- Para os jobs do tipo 1:

$\bar{w} = \bar{d} + \bar{s} = 5.73 + 1.99 = 7.72$
$\bar{l} = \bar{q} + \bar{x} = 0.96 + 0.33 = 1.29$

Além disso

- o tempo médio de serviço está próximo de 2.00
- o tempo médio entre chegada é 2.38 que está próximo de 2.40