Teoría de Colas

Rudik Rompich y David Cuellar

Departamento de Computación, Modelación y Simulación,
Universidad del Valle de Guatemala, Ciudad de Guatemala, Guatemala

rom19857@uvg.edu.gt, cue18382@uvg.edu.gt

***Resumen*—Las líneas o colas de espera comprenden un espectro cotidiano en la vida de las personas; las cuales se pueden encontrar en las compras del supermercado, los trámites burocráticos, la parada del autobús, la entrada a conciertos o discotecas, etcétera. En este proyecto se determinaron los resultados de dos problemas: (1) Los tiempos de salida basado en diferentes arreglos y basado en dos listas de tiempos de llegada y de salida respectivamente. (2) La comprobación y formulación de ecuaciones para determinar la igualdad entre la resta del tiempo de salida y el tiempo de servicio siendo directamente proporcional al máximo entre la hora de llegada y una lista de horas de salida basado en el número de servidores.**

I. INTRODUCCIÓN

La resolución de los problemas propuestos se analizaron desde las perspectivas teóricas y prácticas. La metodología usada se basó en: (1) Creación de algoritmos de simulación. (2) Determinación y comprobación de las ecuaciones a través de la experimentación directa con un servidor y con dos servidores independientes. El desarrollo del proyecto consistió en el uso de un lenguaje de programación (*python*) y un programa de hojas de cálculo (*excel*). Finalmente, en el proyecto se comprobó y se determinó una ecuación que puede ser generalizada a k servidores en una línea de espera.

II. DESARROLLO DE CONTENIDOS

II-A. Problemas 1.a, 1.b y 2.d

La resolución de esta parte se llevó a cabo totalmente en *Python*. El problema otorga trece datos de tiempos de llegada y de servicio.

```
ensayos = 13
tiempo_de_llegadas =
  [12,31,63,95,99,154,198,221,304,346,411,455,537]
tiempo_de_servicio =
  [40,32,55,48,18,50,47,18,28,54,40,72,12]
```

Listing 1. Tiempos de llegada y de servicio.

Ahora bien, nótese que los tiempos de llegada de listing 1 presentan los tiempos ya ordenados. Sin embargo, el algoritmo desarrollado exige que únicamente se den los tiempos entre llegadas en bruto. Las distribuciones que pueden ser utilizadas en el algoritmo pueden ser encontradas en [1]. A continuación se presenta el algoritmo para quitarles el orden a los tiempos de llegada.

```
# Casos
caso_a_b= [tiempo_de_llegadas[0]]
caso_c = []
for i in range(1,len(tiempo_de_llegadas)):
  elemento =  tiempo_de_llegadas[i] -
  tiempo_de_llegadas[i-1]
  caso_a_b.append(elemento)
  caso_c.append(elemento)
caso_c.sort(reverse=False)
caso_c.insert(0,tiempo_de_llegadas[0])
```

Listing 2. Casos tiempos de llegada

Para modelar el primero de los casos, únicamente se asigna la variable para los tiempos entre llegadas.

```
tiempo_entre_llegada= caso_a_b
```

Listing 3. Caso a y Caso b

Posteriormente, se modelaron los dos casos requeridos: un servidor y dos servidores. Por motivos didácticos, únicamente se muestra el caso para dos servidores; para el caso de un servidor la solución es análoga y puede consultarse en el apéndice el código. Las variables utilizadas son las siguientes:

$$
\begin{aligned}
T_n^e &= \text{Tiempo entre llegadas}\\
A_n &= \text{Tiempo de llegada}\\
T_n^c &= \text{Tiempo comienzo de servicio}\\
T_n^w &= \text{Tiempo de espera}\\
S_n &= \text{Tiempo de servicio}\\
D_n &= \text{Tiempo de completación}\\
T_n^s &= \text{Tiempo en el sistema}\\
C_n^1 &= \text{Caja 1}\\
C_n^2 &= \text{Caja 2}
\end{aligned}
$$

Por otra parte, las ecuaciones planteadas son las siguientes:

$$A_1 = T_1^e = T_1^c \tag{1}$$
$$C_1^1 = D_1 \tag{2}$$
$$C_1^2 = 0 \tag{3}$$

$$A_n = A_{n-1} + T_n^e, n > 1 \tag{4}$$

$$T_n^e = \begin{cases} \text{mín}\{C_{n-1}^1, C_{n-2}^2\}, & A_n \leq \text{mín}\{C_{n-1}^1, C_{n-1}^2\} \\ A_n, & A_n > \text{mín}\{C_{n-1}^1, C_{n-1}^2\} \end{cases}, n > 1 \tag{5}$$

$$T_n^w = T_n - A_n, n \geq 1 \tag{6}$$

$$D_n = T_n^c + S_n, n \geq 1 \tag{7}$$

$$T_n^s = D_n - A_n, n \geq 1 \tag{8}$$

$$C_n^1 = \begin{cases} D_n, & C_{n-1}^1 = \text{mín}\{C_{n-1}^1, C_{n-1}^2\} \\ C_{n-1}^1, & C_{n-1}^1 \neq \text{mín}\{C_{n-1}^1, C_{n-1}^2\} \end{cases}, n > 1 \tag{9}$$

$$C_n^2 = \begin{cases} D_n, & C_{n-1}^2 = \text{mín}\{C_{n-1}^1, C_{n-1}^2\} \\ C_{n-1}^2, & C_{n-1}^2 \neq \text{mín}\{C_{n-1}^1, C_{n-1}^2\} \end{cases}, n > 1 \tag{10}$$

De dichas ecuaciones se genera un *dataframe* que proporciona los tiempos de salida requeridos para los primeros dos casos.

II-B. Problema 1.c

Análogamente, para el caso c únicamente es necesario repetir el procedimiento del inciso anterior modificado esta única línea de código.

```
tiempo_entre_llegada= caso_c
```

Listing 4. Caso c

II-C. Problema 2.a, 2.b y 2.c

Las ecuaciones correspondientes se determinaron a partir de la generalización del caso de un único servidor dado para $D_0 = 0$ y para $n > 0$ tal que

$$D_n - S_n = \text{máx}\{A_n, D_{n-1}\} \tag{11}$$

III. Resultados

IV. Conclusiones y discusión

1. a
2. a
3. a
4. a
5. a
6. a
7. a
8. a
9. a
10. a
11. a
12. a
13. a
14. a
15. a
16. a
17. a
18. a
19. a
20. a
21. a
22. a
23. a

IV-A. Subsection Heading Here

IV-A1. Subsubsection Heading Here:

V. Conclusion

Apéndice A
Proof of the First Zonklar Equation

Agradecimientos y Reconocimientos

Referencias

[1] Dennis Wackerly, William Mendenhall, and Richard L Scheaffer. *Mathematical statistics with applications*. Cengage Learning, 2014.