

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.

```
1 ensayos = 13
2 tiempo_de_llegadas =
  [12, 31, 63, 95, 99, 154, 198, 221, 304, 346, 411, 455, 537]
3 tiempo_de_servicio_a_b =
  [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 [2]. A continuación se presenta el algoritmo para quitarles el orden a los tiempos de llegada.

```
1 caso_a_b= [tiempo_de_llegadas[0]]
2
```

```
for i in range(1, len(tiempo_de_llegadas)):
    elemento = tiempo_de_llegadas[i] -
    tiempo_de_llegadas[i-1]
    caso_a_b.append(elemento)
```

Listing 2. Casos tiempos de llegada

El código está basado para modelar cualquier tipo de distribución, por lo tanto, para este caso únicamente es necesario modificar estas dos variables.

```
1 tiempo_entre_llegada = caso_a_b
2 tiempo_de_servicio = tiempo_de_servicio_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:

T_n^e = Tiempo entre llegadas
 A_n = Tiempo de llegada
 T_n^c = Tiempo comienzo de servicio
 T_n^w = Tiempo de espera
 S_n = Tiempo de servicio
 D_n = Tiempo de completación
 T_n^s = Tiempo en el sistema
 C_n^1 = Caja 1
 C_n^2 = Caja 2

Por otra parte, las ecuaciones planteadas son las siguientes:

$$A_1 = T_1^e = T_1^c \quad (1)$$

$$C_1^1 = D_1 \quad (2)$$

$$C_1^2 = 0 \quad (3)$$

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

$$T_n^e = \begin{cases} \min\{C_{n-1}^1, C_{n-2}^2\}, & A_n \leq \min\{C_{n-1}^1, C_{n-1}^2\} \\ A_n, & A_n > \min\{C_{n-1}^1, C_{n-1}^2\} \end{cases}, n > 1 \quad (5)$$

$$T_n^w = T_n - A_n, n \geq 1 \quad (6)$$

$$D_n = T_n^c + S_n, n \geq 1 \quad (7)$$

$$T_n^s = D_n - A_n, n \geq 1 \quad (8)$$

$$C_n^1 = \begin{cases} D_n, & C_{n-1}^1 = \min\{C_{n-1}^1, C_{n-1}^2\} \\ C_{n-1}^1, & C_{n-1}^1 \neq \min\{C_{n-1}^1, C_{n-1}^2\} \end{cases}, n > 1 \quad (9)$$

$$C_n^2 = \begin{cases} D_n, & C_{n-1}^2 = \min\{C_{n-1}^1, C_{n-1}^2\} \\ C_{n-1}^2, & C_{n-1}^2 \neq \min\{C_{n-1}^1, C_{n-1}^2\} \end{cases}, n > 1 \quad (10)$$

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 el código para adaptarse a una línea de espera con modalidad *LIFO* (último en entrar y primero en salir, por sus siglas en inglés); la parte teórica se puede encontrar ampliamente en la literatura [1]. En el caso particular del proyecto, este inciso únicamente requirió realizar la simulación manualmente.

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

$$D_n - S_n = \max\{A_n, D_{n-1}\} \quad (11)$$

La demostración práctica para el caso de $k = 1$ puede ser fácilmente deducida con una hoja de cálculo; mientras que para la generalización $k > 1$ simplemente es necesario hacer una modificación al segundo término de la expresión del máx. Se ofrecen comprobaciones de los casos $k = 1$ y $k = 2$ en este repositorio; de donde se demuestra intuitivamente el caso general.

III. RESULTADOS

III-A. Problemas 1.a, 1.b, 1.c y 2.d

Los tiempos de finalización para los distintos tipos de arreglos pueden ser observados en el table I.

Cuadro I
TIEMPOS DE FINALIZACIÓN

Problema 1.a	Problema 1.b	Problema 1.c
52	52	52
84	63	84
139	118	139
187	143	157
205	136	207
255	204	254
302	245	272
320	239	320
348	332	348
402	400	402
451	451	451
527	527	527
549	549	549

III-B. Problemas 1.a, 1.b, 1.c y 2.d

Se determinó que el caso con un único servidor ($k = 1$) sí se cumple dado $D_0 = 0$ y para $n > 0$ tal que

$$D_n - S_n = \max\{A_n, D_{n-1}\} \quad (12)$$

Mientras que para dos servidores ($k = 2$) se cumple para $D_0 = D_{-1} = 0$ y para $n > 2$ tal que

$$D_n - S_n = \max\{A_n, \min\{D_{n-1}, D_{n-2}\}\} \quad (13)$$

Finalmente, el caso general puede ser expresado para k servidores, en donde se cumple si $D_0 = D_1 = \dots = D_{-(k-1)} = 0$ tal que,

$$D_n - S_n = \max\{A_n, \min\{D_{n-1}, D_{n-2}, \dots, D_{n-k}\}\} \quad (14)$$

IV. CONCLUSIONES Y DISCUSIÓN

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.

Se comprobó dos modelos de líneas de espera que se enfocan en sus contrapartes teóricas: *LIFO* (último en entrar y primero en salir) y *FIFO* (primero en entrar y primero en salir). Naturalmente, el modelo *FIFO* [1] representa el modelo usual que se usa en todo tipo de línea de espera en la vida diaria, mientras que el modelo *LIFO* está relegado a otro tipo de aplicaciones más complejas. Se determinó, véase table I, que no hay distinción en el uso de modelos *LIFO* y *FIFO* ni el uso de uno o dos servidores; ya que todos hicieron un uso de 549 unidades de espera. Esto podría estar directamente relacionado a las distribuciones específicas que se dieron; que no representan probabilidades aleatorias sino distribuciones arbitrariamente dadas. Por otra parte, se determinó una propiedad en la función mínimo de la eq. (14) la cual hace uso

necesario de la propiedad $D_0 = D_1 = \dots = D_{-(k-1)} = 0$ para que dicha función no presente ninguna incongruencia.

Finalmente, para futuras investigaciones se recomienda asignar distribuciones de probabilidades aleatorias para una mejor representación del uso de 1 o más cajeros; además se podría intentar investigar nuevas propiedades y ecuaciones que puedan ser útiles para generalizar nuevos resultados en las líneas de espera.

Se concluye

1. Se determinó que las líneas de espera se constituyen principalmente basados en el modelo *FIFO* que son las más justas y efectivas para ordenar dichas líneas.
2. Se observó que las distribuciones de probabilidad aleatorias pueden representar una forma más efectiva para poner a prueba los modelos de simulación de varios cajeros.
3. Se investigó diferentes modelos de simulación en donde no es necesario caer a la simulación directamente para las líneas de espera; ya existen modelos teóricos que calculan probabilidades, cantidades y otros aspectos con un par de simples fórmulas.

APÉNDICES

IV-A. Código utilizado

En este repositorio.

IV-B. Hojas de cálculo

En este repositorio.

REFERENCIAS

- [1] S.M. Ross. *Simulation*. Knovel Library. Elsevier Science, 2013.
- [2] Dennis Wackerly, William Mendenhall, and Richard L Scheaffer. *Mathematical statistics with applications*. Cengage Learning, 2014.