



Simulación de Eventos Discretos

Claudia Alvarez Martínez

Roger Moreno Gutiérrez

27 de febrero de 2024

1 Problema de Reparaciones

Un sistema necesita n máquinas funcionando para estar operativo. Para protegerse contra las averías de las máquinas, se mantienen máquinas adicionales disponibles como repuestos. Cuando una máquina se avería, es inmediatamente reemplazada por un repuesto y se envía a la instalación de reparación, que consta de un único reparador que repara las máquinas averiadas una a la vez. Una vez que una máquina averiada ha sido reparada, se vuelve a disponer como repuesto para ser utilizada cuando sea necesario. Todos los tiempos de reparación son variables aleatorias independientes que tienen la función de distribución común G . Cada vez que una máquina se pone en uso, la cantidad de tiempo que funciona antes de averiarse es una variable aleatoria, independiente del pasado, que tiene la función de distribución F . Se dice que el sistema falla cuando una máquina falla y no hay repuestos disponibles. Asumiendo que inicialmente hay $n + s$ máquinas funcionales de las cuales n se ponen en uso y s se mantienen como repuestos, estamos interesados en simular este sistema para aproximar $E[T]$, donde T es el tiempo en el que el sistema se estrella.

2 Introducción

El proyecto aborda un escenario donde un sistema depende de un número específico de máquinas para estar operativo. Para mitigar los impactos de posibles averías, se tienen máquinas adicionales como repuestos. Cuando una máquina falla, es reemplazada por un repuesto y se envía a reparación.

Las variables clave en el sistema incluyen:

- Tiempo de Reparación (G): Se modela como una variable aleatoria con función de distribución común G .
- Tiempo de Funcionamiento antes de Averiarse (F): Otra variable aleatoria independiente con función de distribución F .

El objetivo principal es simular el sistema y estimar el tiempo esperado hasta que el sistema experimente una falla completa, definido como el tiempo en el cual no hay repuestos disponibles cuando una máquina falla.

Para la estimación de la durabilidad de una máquina se utiliza (en el ejemplo ofrecido) una variable aleatoria exponencial. Dicho esto, las variables de entrada del problema son:

- n : Cantidad de máquinas que funcionan a la vez.
- s : Cantidad de repuestos disponibles.

3 Detalles de implementación

Se define una clase principal **RepairSystem** encargada de hacer la simulación. Su constructor toma las variables de entrada, e inicializa otras variables del problema. Para ello tendremos:

- t : Tiempo ocurrido en la simulación.
- t_{star} : Tiempo en el que la máquina actualmente en reparación estará operativa (∞ si no hay ninguna en reparación).
- r : Número de máquinas fuera de servicio en un momento dado.

Teniendo en cuenta la distribución especificada F , se precalcula el tiempo de ruptura de las máquinas operativas, almacenando sus valores en un **Heap**, definido en el módulo *heap.py*, lo que permite obtener eficientemente en todo momento el mínimo de este conjunto, lo que equivale al tiempo en que falla la primera máquina.

La variable de estado r cambiará en dos casos, los que a su vez definen los posibles eventos del sistema:

1. Si se rompe una nueva máquina operativa.
2. Si una máquina acaba de ser reparada.

La función **simulateSystem()** es la encargada de iterar por los eventos de la simulación. Sea t_1 en tiempo en que se romperá la próxima máquina, es importante saber si se rompe una máquina en funcionamiento ($t_1 < t_{star}$), o si se arregla una de la cola de reparación ($t_{star} \leq t_1$). En ambos casos se actualiza la variable de estado, y luego, en el primer caso:

- Se comprueba si el sistema ha fallado, para tal caso se recopilan los datos relevantes de la simulación (el tiempo transcurrido).
- Si la cantidad de máquinas en reparación es como mucho la cantidad de repuestos, el sistema no ha fallado, por lo que se reemplaza la máquina rota por un repuesto, estimando su tiempo de vida con F .
- Si la máquina que acaba de romperse es la única rota en el tiempo t actual se inicia su reparación, estimando su tiempo final con G .

En el segundo caso:

- Se genera el tiempo de reparación de la siguiente máquina en la cola con G .
- En caso de que la máquina arreglada sea la última de la cola, establecer en tiempo de reparación de la siguiente en ∞ .

4 Resultados y Experimentos

Tras k ejecuciones de la simulación se recopilan los tiempos de parada de estas. Al ser variables aleatorias independientes que representan el tiempo de falla de la simulación, con el promedio de estos datos podemos estimar el tiempo esperado de falla del sistema, y esto es $E[T]$.

Surge la pregunta de cuándo detener la simulación, para ello hemos utilizado un método que estima el valor de k cuando tras 100 iteraciones no se alcanzan umbrales aceptables de desviación. La idea es continuar generando variables aleatorias de la simulación hasta cumplir tal condición.

$$S/\sqrt{k} < d$$

Donde S es la desviación estándar, k el número de iteraciones y d el umbral definido.

En el archivo **results.txt** se encuentra una recopilación de datos tras 1000 simulaciones independientes para diferentes valores de las variables de entrada. Basándonos en estos resultados, llegamos a las siguientes conclusiones:

- El tiempo de falla de la simulación tiende a ser mayor cuando la proporción n/s se acerca a 0, por otro lado, mientras n/s tiende a ser ∞ , el sistema se estrella rápidamente. Esto se entiende fácilmente, pues no hay los repuestos suficientes para reemplazar las máquinas rotas.
- Tras un análisis de la desviación estándar de los tiempos de fallo, se observa que cuando la proporción n/s tiende a 0, hay una mayor dispersión de los datos. Ocurre de forma contraria cuando esta proporción tiende a ∞ .

El análisis estadístico resulta necesario, esto se puede ver en la estimación del valor de k , pues permite obtener resultados más precisos de la simulación, al hacer que esta se ejecute la cantidad de veces necesarias para lograr reducir la posible desviación desproporcionada, producto de una cantidad escasa de iteraciones, en algunos casos. Además, comparar estadísticas simuladas permite evaluar qué tan bien se ajusta el modelo a la realidad. Esto garantiza de cierto modo la confiabilidad de las conclusiones derivadas de la simulación.