Modelos y Simulación

Trabajo Especial I

Modelo de Reparación

Giovanni Rescia

FaMAF, UNC

21 de Mayo de 2014

1 Resumen

El objetivo de este trabajo es poder estimar mediante simulaciones, la vida útil de un lavadero con N máquinas en funcionamiento, S máquinas de repuesto y K operarios que reparan las máquinas que se van rompiendo. La vida útil estará representada por ”cuánto tiempo funciona el sistema hasta que hay menos de N máquinas en funcionamiento” teniendo en cuenta las de repuesto y aquellas que vayan reparando a lo largo del tiempo los operarios. Y también poder responder a preguntas del tipo ”para mejorar la media del tiempo antes que el sistema falle, conviene agregar un operario o una máquina?”.

2 Introducción

Un lavadero de ropa automático, cuenta con N máquinas lavadoras en servicio y con S máquinas de repuesto, todas ellas de idéntica marca, modelo y antigüedad. Además el lavadero cuenta con los servicios de un técnico que repara las máquinas. Obviamente, el técnico repara las máquinas en serie, encargándose de una sola por vez. El problema consiste en determinar el tiempo medio (y su correspondiente desviación estándar) que transcurre hasta que el lavadero deja de ser operativo (fallo del sistema), esto es, el momento en el que se tiene menos de N máquinas en servicio, o lo que es lo mismo, posee mas de S máquinas defectuosas en el taller.

Todos los tiempos de funcionamiento de las maquinas hasta descomponerse son variables independientes exponenciales con un tiempo medio hasta fallar de T_F , y que el tiempo de reparación de una máquina que ingresa a taller es una variable exponencial con tiempo medio igual a T_R , independiente de todos los anteriores.

2.1 Bases teóricas para la resolución del problema

A continuación se presentan las bases teóricas en las cuales este modelo está fuertemente basado para su resolución

2.1.1 Cálculo de la Esperanza y la Varianza por la ley fuerte de los grandes números

Esperanza

Teorema: Sea X_1,X_2,...,X_n una sucesión de variables aleatorias independientes e idénticamente distribuidas con media μ, entonces vale que

$$\lim_{n \to \infty} \frac{X_1 + X_2 + ... + X_n}{n} = \mu$$

Basándonos en ese resultado, podemos obtener al correr el algoritmo n veces para obtener la vida útil del sistema y al dividirlo por n obtendremos la esperanza (con n suficientemente grande).

Varianza

Para el cálculo de σ^2 utilizaremos el siguiente estimador insesgado (i.e, sea θ una v.a. entonces $\hat{\theta}$ es un estimador insesgado de θ si $E[\hat{\theta}]$=θ). Antes de dar un estimador insesgado de la varianza, daremos uno de la esperanza, ya que la varianza depende de la esperanza.

Proposición: Sea X_1,X_2,...,X_n una sucesión de variables aleatorias independientes e idénticamente distribuidas con media μ, y sea

$$\bar{X} = \sum_{i=1}^{n} \frac{X_i}{n}$$

entonces vale que

$$E[\bar{X}] = \mu$$

Ahora que tenemos un estimador insesgado para μ, tenemos la siguiente proposición:

Proposición: Sea X_1,X_2,...,X_n una sucesión de variables aleatorias independientes e idénticamente distribuidas con esperanza μ y varianza σ^2 y sea $\bar{X}$ un estimador insesgado de μ entonces si

$$S^2 = \sum_{i=1}^{n} \frac{(X_i - \bar{X})^2}{n-1}$$

entonces S^2 es un estimador insesgado para σ^2. Por lo que la desviación estándar σ es igual a S.

2.1.2 Generación de variables aleatorias uniformes continuas en el (0,1)

Contamos con un método eficiente para generar una variable aleatoria $U \sim \mathcal{U}(0,1)$.

Se podría considerar como la parte algorítmica más importante, ya que nos va a permitir generar variables aleatorias exponenciales. (Se puede generar cualquier variable aleatoria solo que en este problema solo nos interesan las exponenciales).

2.1.3 Tiempos de fallo y reparación de las máquinas

Trabajamos bajo las siguientes hipótesis:

- Todos los tiempos de funcionamiento de las máquinas hasta descomponerse son variables independientes exponenciales con un tiempo medio hasta fallar de 1 mes. Es decir, el tiempo de funcionamiento hasta descomponerse es una variable aleatoria $\sim \varepsilon(1)$
- El tiempo de reparación de una máquina que ingresa a taller es una variable exponencial con tiempo medio igual a 1/8 de mes, independiente de todos los anteriores. Es decir, el tiempo de reparación de una máquina es una variable aleatoria $\sim \varepsilon(8)$

2.1.4 Generación de variables aleatorias exponenciales

Gracias a que podemos generar una variable aleatoria $U \sim \mathcal{U}(0,1)$ podemos implementar un algoritmo para generar variables aleatorias $\sim \varepsilon(\lambda)$.

El pseudo código a continuación:

<u>Generar $X \sim \varepsilon(\lambda)$:</u>

1. Generar $U \sim \mathcal{U}(0,1)$
2. $X \leftarrow -\frac{1}{\lambda} \log(U)$

Entonces los algoritmos para generar los tiempos de vida hasta que se descompone y hasta que se arregla una máquina están dados por:

<u>Generar tiempo hasta descomponerse $X \sim \varepsilon(1)$:</u>

1. Generar $U \sim \mathcal{U}(0,1)$
2. $X \leftarrow -log(U)$

<u>Generar tiempo hasta arreglo $X \sim \varepsilon(8)$:</u>

1. Generar $U \sim \mathcal{U}(0,1)$
2. $X \leftarrow -\frac{1}{8} \log(U)$

3 Pseudo Código para realizar la simulación

(Nota: por cuestiones de espacio, el algoritmo fue separado en 3 partes: la primera tiene las variables y sus respectivas inicializaciones, la segunda parte describe el evento en que se rompe un máquina antes que otra sea arreglada, y la tercer parte describe el evento en que se repara una máquina antes que otra falle)

Algorithm 1 Simuación con n máquinas en funcionamiento, s de repuesto y un operario

procedure SIMULACIÓN
    ▷ Inicialización de variables
    ▷ Cantidad de máquinas en funcionamiento para que el sistema no falle
    $working\_machines \leftarrow n$
    ▷ Máquinas de repuesto
    $spare\_machines \leftarrow s$
    $system\_failure \leftarrow False$
    ▷ Tiempo final en meses (output)
    $T \leftarrow 0$
    ▷ Variable de tiempo
    $t \leftarrow 0$
    ▷ $broken\_machines$ va a marcar el estado del sistema; son las máquinas rotas en el tiempo t. Esta variable va a cambiar cuando una máquina falle o una máquina se repare
    $broken\_machines \leftarrow 0$
    ▷ Tiempo en que se reparó la última máquina
    $machine\_repair\_time \leftarrow \infty$
    ▷ Guardo los tiempos de falla de las máquinas en un conjunto, por ende, está ordenado
    $machines\_failure\_time \leftarrow \emptyset$
    ▷ Generamos los tiempos de falla de las máquinas
    for $i \leftarrow 1$ **to** n **do**
        $X \leftarrow \varepsilon(1)$
        $machines\_failure\_time \leftarrow machines\_failure\_time \cup \{X\}$
    end for

Algorithm 1 Simuación con n máquinas en funcionamiento, s de repuesto y un operario(continuación)

▷ Mientras hallan menos máquinas rotas que de repuesto, el sistema va a funcionar normalmente
while not $system\_failure$ **do**
▷ Nos fijamos en la primer máquina en fallar
$t_1 \leftarrow$ *Menor elemento (tiempo) del conjunto*
▷ Evento de tipo 1: falló una máquina antes que otra se arregle
if $t_1 < machine\_repair\_time$ **then**
▷ Actualizamos el tiempo t
$t \leftarrow t_1$
▷ Tenemos otra máquina rota, pues falló antes que se arrgle otra
$broken\_machines \leftarrow broken\_machines + 1$
▷ Si hay más máquina en reparación que de repuesto, entonces el sistema falla, pues no me alcanza para tener n en funcionamiento
if $broken\_machines > spare\_machines$ **then**
▷ T representa el tiempo total que el sistema estuvo en funcionamiento
$T \leftarrow t$
▷ El sistema ha fallado
$system\_failure \leftarrow True$
▷ Cuento con máquinas de repuesto, el sistema se mantiene estable
else
▷ Genero el tiempo que va a durar una de las máquinas de repuesto que pongo en funcionamiento
$X \leftarrow \varepsilon(1)$
▷ Ya no necesito la primer máquina que falló, pues es reemplazada
$machines\_failure\_time \leftarrow machines\_failure\_time - \{t_1\}$
▷ La máquina de repuesto va a entrar en funcionamiento en el momento t, y va durar X, por lo que la falla se producirá en el tiempo $X + t$
$machines\_failure\_time \leftarrow machines\_failure\_time \cup \{X + t\}$
end if
▷ Si hay una máquina esperando ser reparada, la reparo
if $broken\_machines == 1$ **then**
▷ Genero un tiempo de reparación de la máquina
$Y \leftarrow \varepsilon(8)$
▷ La máquina se va a empezar a reparar en el tiempo t, y la reparación va a tomar un tiempo Y, por lo tanto, la reparación se va a completar en el tiempo $Y + t$
$machine\_repair\_time \leftarrow Y + t$
end if

Algorithm 1 Simuación con n máquinas en funcionamiento, s de repuesto y un operario(continuación)

▷ Evento de tipo 2: una máquina es reparada antes que falle una que está en funcionamiento
else
▷ Actualizamos el tiempo t
$t \leftarrow machine\_repair\_time$
▷ Reparé una máquina, por lo tanto tengo una máquina rota menos
$broken\_machines \leftarrow broken\_machines - 1$
▷ Si todavía hay máquinas rotas, reparo una
if $broken\_machines > 0$ **then**
▷ Genero un tiempo de reparación de la máquina
$Y \leftarrow \varepsilon(8)$
▷ La máquina se va a empezar a reparar en el tiempo t, y la reparación va a tomar un tiempo Y, por lo tanto, la reparación se va a completar en el tiempo $Y + t$
$machine\_repair\_time \leftarrow Y + t$
end if
▷ Si no hay máquinas rotas, entonces la próxima máquina va a tomar ∞ tiempo en ser reparada (pues no hay ninguna que reparar en este instante t)
if $broken\_machines == 0$ **then**
▷ Restablezco el tiempo de reparación de la máquina
$machine\_repair\_time \leftarrow \infty$
end if
end if
end while
RETURN T
end procedure

Este algoritmo describe una simulación para n máquinas en funcionamiento, s máquinas de repuesto y un solo operario.

3.1 Algoritmo para un sistema con dos operarios

La lógica del algoritmo es casi la misma, hay que agregarle las siguientes líneas al algoritmo anterior para poder lograr una simulación de un sistema con n máquinas en funcionamiento, s máquinas de repuesto y dos operarios que arreglan máquinas. Las siguiente líneas de pseudo código se embeben justo después de la línea 53 del algoritmo anterior (se van a mostrar las líneas 49 a 53 también para ponerlo en contexto).

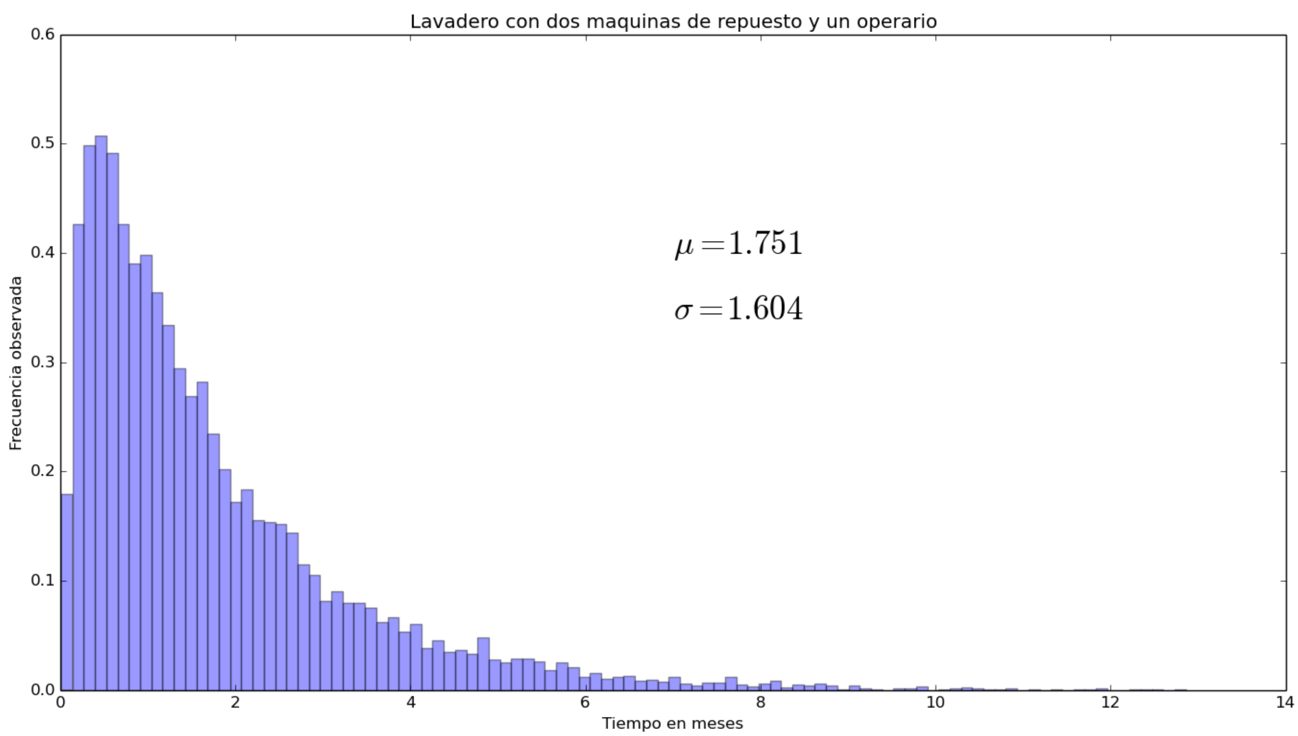
Algorithm 2 Simuación con n máquinas en funcionamiento, s de repuesto y dos operarios

procedure SIMULACIÓN
  if $broken\_machines == 1$ **then**
    ▷ Genero un tiempo de reparación de la máquina
    $Y \leftarrow \varepsilon(8)$
    ▷ La máquina se va a empezar a reparar en el tiempo t, y la reparación va a tomar un tiempo Y, por lo tanto, la reparación se va a completar en el tiempo $Y + t$
    $machine\_repair\_time \leftarrow Y + t$
  end if
  ▷ Si hay dos máquinas en reparación, es porque otra se rompió antes que la última que entró en reparación se halla reparado. Como tengo otro operario más, la puedo poner a reparar
  if $broken\_machines == 2$ **then**
    ▷ Genero un tiempo de reparación de la máquina
    $Y \leftarrow \varepsilon(8)$
    ▷ La máquina se va a empezar a reparar en el tiempo t, y la reparación va a tomar un tiempo Y, por lo tanto, la reparación se va a completar en el tiempo $Y + t$
    $machine\_2\_repair\_time \leftarrow Y + t$
    ▷ Como los eventos que observo son la reparación y el fallo de una máquina, lo que me va a interesar obtener en este caso es la máquina que sea reparada en menor tiempo entre la máquina que estaba siendo reparada y la que acaba de entrar (en el tiempo t) a ser reparada
    $machine\_repair\_time \leftarrow \min\{machine\_repair\_time, machine\_2\_repair\_time\}$
  end if
end procedure

4 Resultados obtenidos

A continuación se muestran los resultados de la simulación. Para cada caso la Simuación se corrió 100000 veces para calcular la Esperanza, tanto como la Varianza (por ende la desviación estándar) y tanto como para obtener puntos para realizar los gráficos.

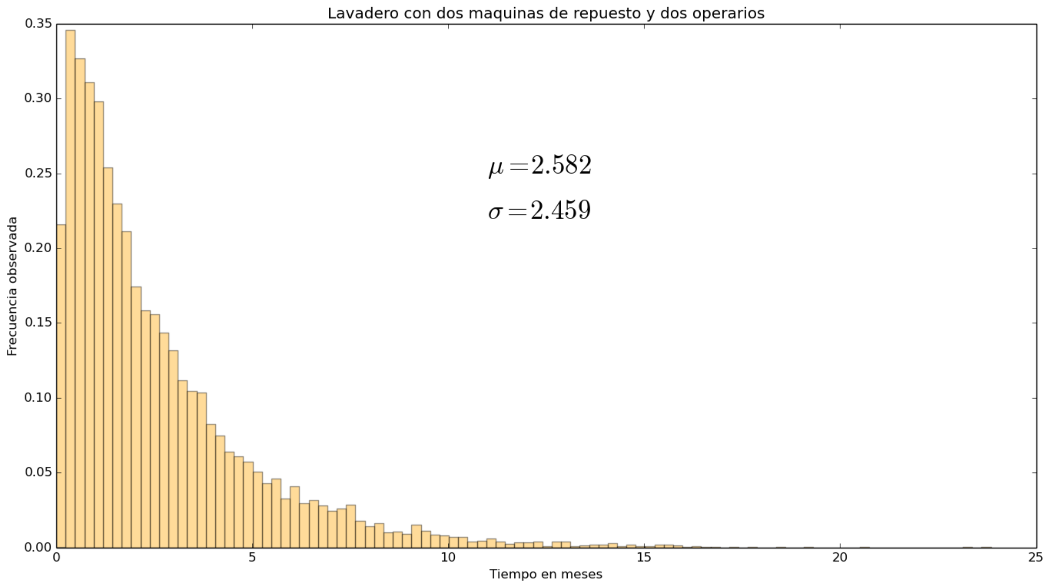
4.1 Resultados con 5 máquinas en funcionamiento, 2 de repuesto y un operario



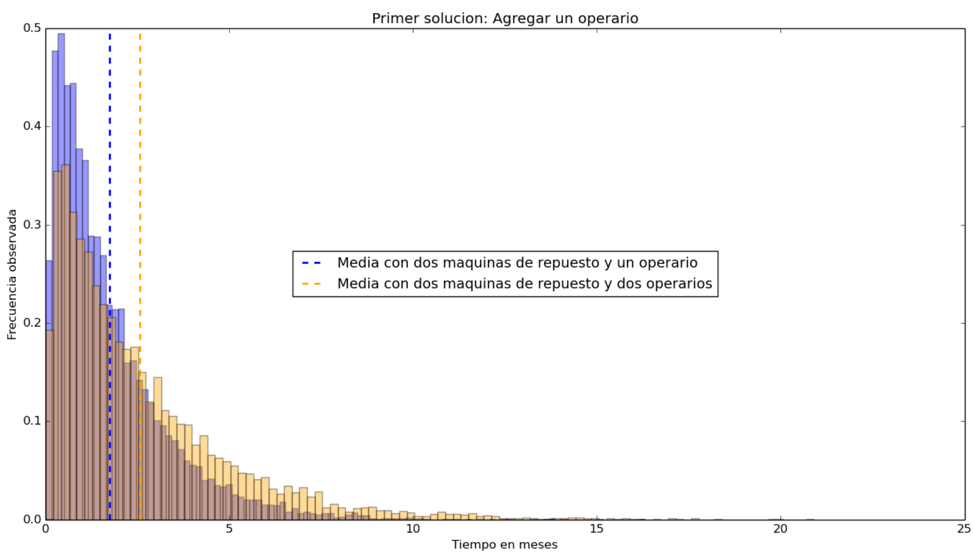
Ahora, si uno quisiera mejorar el tiempo antes que el sistema falle (con 5 máquinas en funcionamiento), podría elegir entre dos opciones: agregar un operario, o agregar una máquina de repuesto.

A continuación, resultados de la simulación para las dos posibles soluciones.

4.2 Solución 1: Agregar un operario

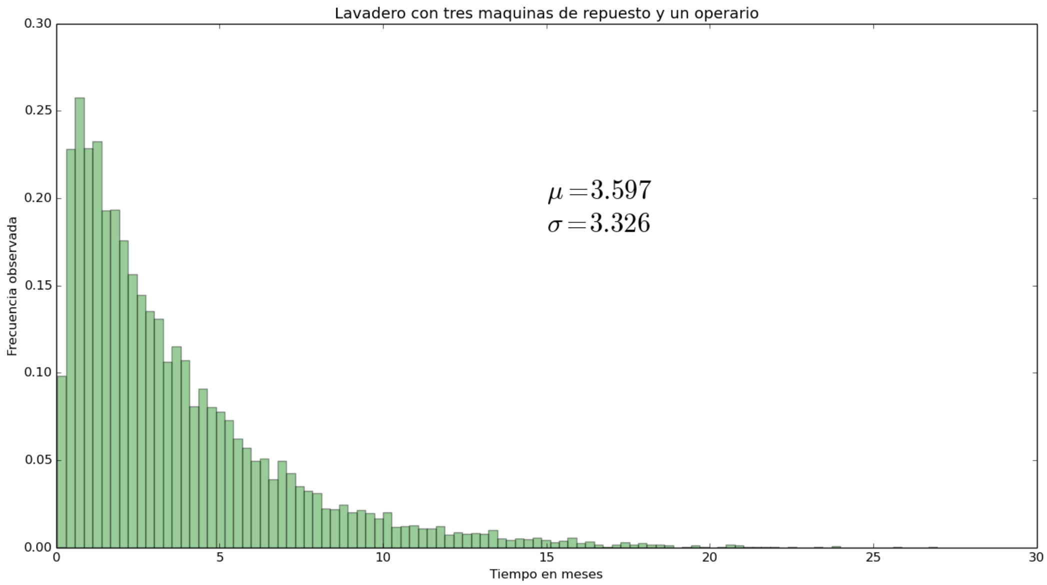


Comparemos cómo ha cambiado el desempeño del sistema:

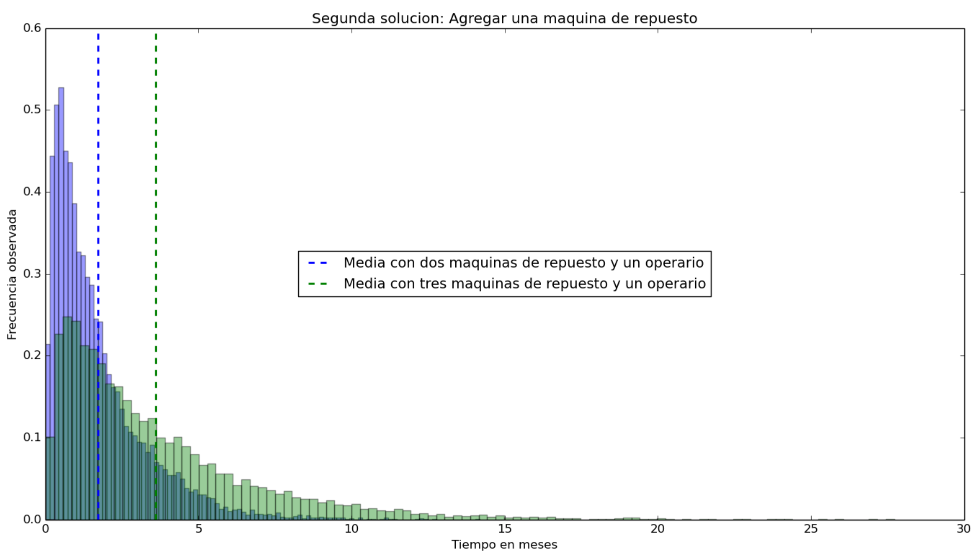


Se puede observar una mejoría de aproximadamente 1 mes en el tiempo medio de falla del sistema.

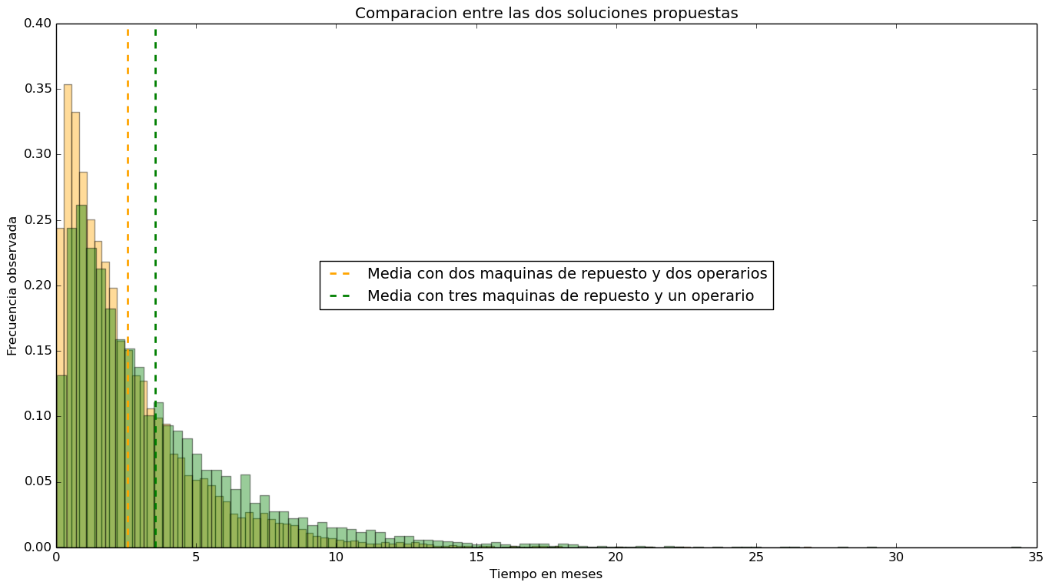
4.3 Solución 2: Agregar una máquina de repuesto



Comparemos cómo ha cambiado el desempeño del sistema:

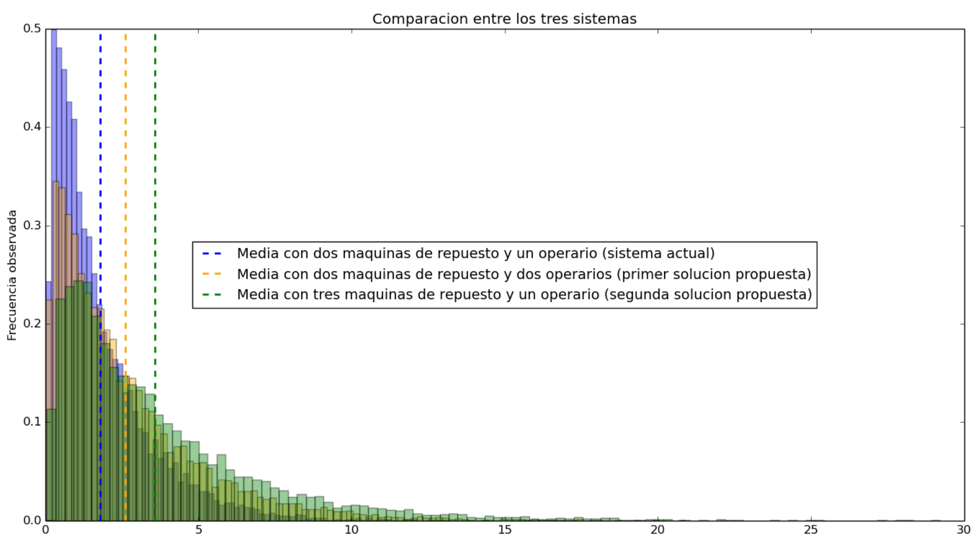


4.4 Comparación entre las soluciones



Podemos notar que el sistema tiene mejor desempeño si agregamos una máquina en vez de un operario.

Comparación entre los tres sistemas:



Tablas con las simulaciones

- **5 máquinas funcionando, 2 de repuesto y un operario**

N	μ	σ^2	σ
100	1.605	2.573	1.545
1000	1.761	2.237	1.652
10000	1.772	2.453	1.616

- **5 máquinas funcionando, 2 de repuesto y dos operarios**

N	μ	σ^2	σ
100	2.651	6.531	2.082
1000	2.652	5.689	2.423
10000	2.601	6.445	2.438

- **5 máquinas funcionando, 3 de repuesto y un operario**

N	μ	σ^2	σ
100	3.695	11.799	2.899
1000	3.517	10.56	3.397
10000	3.628	10.813	3.356

5 Conclusión

Dados los resultados observados podemos ver que contando con ciertas bases teóricas y un poco de información del problema que estamos modelando, podemos ser capaces de responder a incógnitas que parecen muy difíciles de estimar. Incluso si supiéramos que un sistema con 5 máquinas funcionando, dos de repuesto y un operario, tuviera una media de tiempo hasta que falla de 1.75 meses, no sería obvio saber si la media de tiempo hasta que el sistema falla mejora agregando un operario o una máquina de repuesto. Gracias a las hipótesis en las que nos basamos, fuimos capaces de observar, mediante simulaciones, que la media de tiempo hasta que el sistema falla aumenta a 3.6 meses si ponemos una máquina más de repuesto; contra 2.6 meses si agregamos un operario.