

Trabalho Simulação de Eventos Discretos

Um problema de reparação

Jaqueline Lamas da Silva

2023

Sumário

Introdução	2
Desenvolvimento	3
Verification of the Simulation Model	5
Referências	6

Introdução

O modelo que estamos considerando é um exemplo clássico de sistemas de reserva e reparo, onde há máquinas em operação e máquinas de reserva que entram em operação quando uma das máquinas principais falha. O objetivo é determinar o tempo até que todas as máquinas (principais e reservas) falhem.

Propriedades Relevantes do Modelo:

- Tempo de Falha: Distribuição $F(x)$.
- Número de Máquinas Principais (n): Máquinas em operação.
- Número de Máquinas de Reserva (s): Máquinas que substituem as principais quando estas falham.
- Tempo de Reparo: Distribuição $G(x)$
- Estados do Sistema: Número de máquinas operacionais e em reparo.
- Evento de Falha e Reparo: Afetam o estado do sistema.

Desenvolvimento

Para desenvolver o algoritmo da simulação as instruções da seção 7.7 foram seguidas.

Na construção foi considerado o seguinte caso:

- $n = 4$ (4 máquinas em operação);
- $s = 3$ (3 máquinas reserva);
- $F(x) = 1 - e^{-x}$ (tempo de falha das máquinas iid a uma exponencial com $\lambda = 1$)
1/dia
- $G(x) = 1 - e^{-2x}$ (tempo de concerto uma exponencial com $\lambda = 2$)

```
MaquinaEstragou<-function(event.list,r,t, t.concerto)
{
  t<-event.list[1]
  r<-r+1 # falhou
  if(r==s+1)
  {
    return(TempoFalha=t)
  }else{
    x<-rexp(1,rate=1) # tempo que a substituta vai funcionar
    event.list<-sort(c(event.list[-1],x+t))
    if(r==1)
    {
      y<-rexp(1,rate=2) # tempo de concerto
      t.concerto<-t+y # quando ela ficou pronta
    }
    if(event.list[1]<t.concerto)
    {
      MaquinaEstragou(event.list,r,t, t.concerto)
    }
    else
    {
      MaquinaPronta(event.list,r,t,t.concerto)
    }
  }
}

MaquinaPronta<-function(event.list, r, t, t.concerto)
{
  t<-t.concerto
```

```
r<-r-1
if(r>0) #tem alguma máquina quebrada?
{
  y<-rexp(1,rate=2)
  t.concerto<-t+y # quando ela vai ficar pronta
}
if(r==0)
{
  t.concerto<-Inf
}
if(event.list[1]<t.concerto)
{
  MaquinaEstragou(event.list,r,t, t.concerto)
}
else
{
  MaquinaPronta(event.list,r,t, t.concerto)
}
}

# Condições iniciais do sistema
n<-4 # quatro máquinas
s<-3 # 3 máquinas reservas
t<-0 # Tempo
r<-0 # máquinas quebradas
t.concerto<-Inf # Quando a próxima máquina quebrada sera concertada

x<-replicate(10000, rexp(n,rate=1))
Tempos<-apply(x,MARGIN=2,sort)
str(Tempos)

##  num [1:4, 1:10000] 0.4 2.168 2.19 2.49 0.453 ...

Tempos[,1]

## [1] 0.3999176 2.1675325 2.1897105 2.4900994

temposDeFalha<-apply(Tempos, MARGIN = 2, FUN = MaquinaEstragou, r=0 , t=0)
str(temposDeFalha)

##  num [1:10000] 2.382 1.202 1.897 1.01 0.898 ...
```

```
mean(temposDeFalha)
```

```
## [1] 1.524673
```

Com a configuração utilizada de exemplo, o tempo em que todas as máquinas falharam foi de aproximadamente um dia e meio. A taxa utilizada para o tempo de falha foi de 1/dia.

Verification of the Simulation Model

Para acompanhar o funcionamento do programa desenvolvido foi aplicado um teste de mesa, no qual anota-se os valores assumidos pelas variáveis no decorrer da execução do programa. A técnica foi importante para identificar um erro de atualização de uma variável na chamada recursiva.

Referências

ROSS, S. M. Simulation. 5th. Ed. London, UK: Academic Press, 2013.