**一个多节点****声纳系统中同步时钟机制的可靠性评估和系统优化问题**

519030910115 李春一

**摘要**

在主从通信系统中，有时会发生系统故障，导致系统的平均寿命/可靠性有限。

节点数量对系统的平均寿命/可靠性起决定性作用，节点过少会导致工作的主从节点数量不足，节点过多可能由于一个节点的故障导致全局崩溃。因此，若想让系统的正常工作时间尽可能长，需要选取合适的节点个数。

针对不同种类的故障及其影响，本文建立了数学模型，将切换器、节点、系统的工作状态视为马尔科夫过程。通过数值模拟方法（包括定步长与变步长两种），验证了不同节点数目下，系统的平均寿命/可靠性。

在本文声纳系统的物理条件下，20个节点平均寿命最长、15个节点可靠性最高。同时，本文分析了A，B两种节点各自的重要性，证明了其中一个切换器的性能进行提升，都会增加整体的平均寿命/可靠性，从而为系统中切换器、节点的设计指明了方向。

**关键词：**系统可靠性 主从通信系统 数值模拟方法 马尔科夫链

目录

[1． 概述 1](#_Toc71998343)

[1.1. 引言 1](#_Toc71998344)

[1.2. 寿命物理模型 1](#_Toc71998345)

[1.3. 优化目标 2](#_Toc71998346)

[2. 算法选择 2](#_Toc71998347)

[2.1. 定步长 2](#_Toc71998348)

[2.2. 变步长 3](#_Toc71998349)

[2.3. 算法合理性说明 3](#_Toc71998350)

[3. 最大可靠性 4](#_Toc71998351)

[4. 最大平均工作寿命 4](#_Toc71998352)

[5. 拓展部分 5](#_Toc71998353)

[5.1. 概述 5](#_Toc71998354)

[5.2. 提升最大限定寿命 5](#_Toc71998355)

[5.3. 提升切换器性能 6](#_Toc71998356)

[6. 总结 6](#_Toc71998357)

[6.1. 当下的优点与不足 6](#_Toc71998358)

[6.2. 优化方向 7](#_Toc71998359)

# 概述

## 引言

在多节点声纳监听系统中，会发生切换器故障，导致系统无法正常工作。然而，水下的雷达系统不便于修复，因此，设法延长系统的寿命至关重要。

声纳系统由以下三级组成：

* 切换器：分为A，B两种
* 节点：共n个，每个节点包含A，B两种切换器
* 系统：n个节点共同组成完整的声纳系统

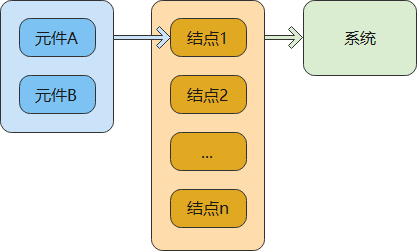


图 1系统组成示意图

## 寿命物理模型

切换器A，B分别有3，2种异常状态，这种工作状态的组合，会产生6种不同的节点状态。

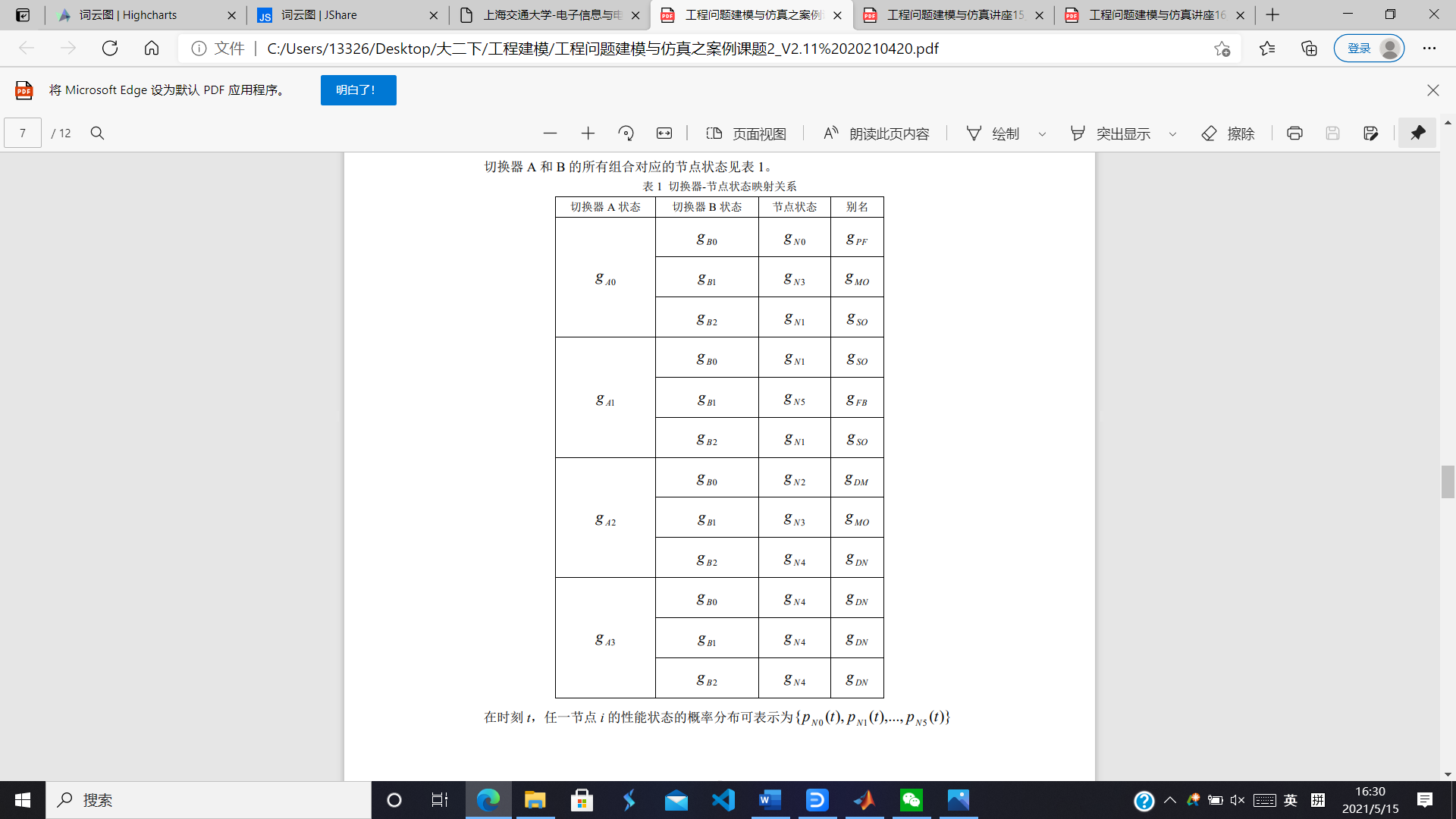


图 2节点状态组合表[1]

* ：节点性能完好
* ：只能作为从节点
* ：或者作为主节点，或者作为不阻塞总线的失效节点
* ：只能作为主节点，否则就会阻塞总线
* ：成为不阻塞总线的失效节点
* ：节点总是阻塞总线

本题中，只有5个节点工作（包含1个主节点），且总线不被阻塞时，系统正常工作。

## 优化目标

* 系统可靠性

若系统在内均可靠工作，则判定系统为可靠。

令系统的工作寿命尽量大于，即进行多次模拟，使得寿命超过的次数尽可能多（概率尽可能大）。

* 系统平均工作寿命

令系统的工作寿命尽可能大，即进行多次模拟，使得平均寿命尽可能大。

# 算法选择

## 定步长

按照以下步骤设计算法：

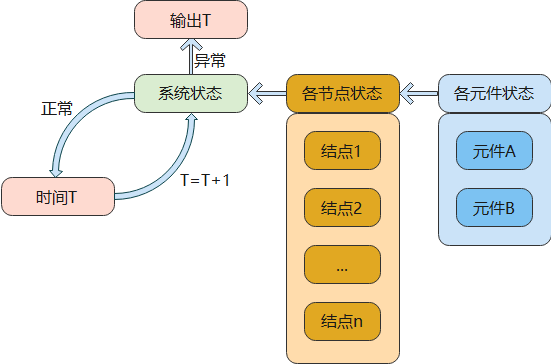


图 3定步长算法示意图

设定步长为，每步改变各元件的状态，改变概率，由此得到各节点的状态，最后模拟出系统的状态。

* 若系统正常，则将寿命增加
* 若系统异常，或总时间大于，输出寿命

此方法可以得出正确结果，但耗时较长。在20个节点的情况下，每尝试100个样本，需要运行。由于使用数值模拟方法，样本数至少为，因此一次计算会耗时接近，因此需要探索速度更快的方法。

## 变步长

按照以下步骤设计算法：

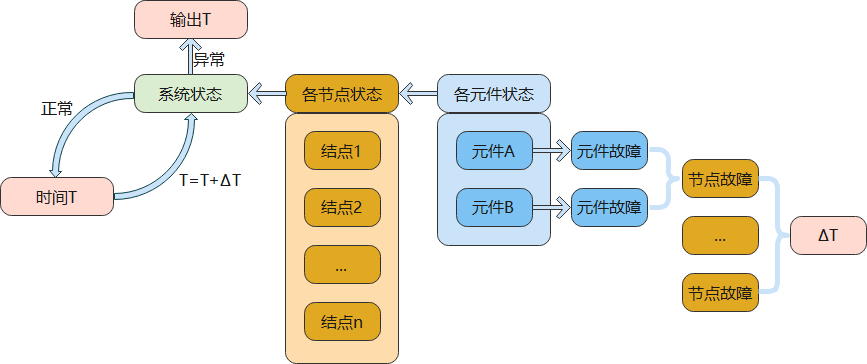


图 4变步长算法示意图

在个节点中，依次模拟个切换器A，B的状态，得到各个切换器发生故障的时间

，，将最短的时间记为，从而得到最先发生故障的切换器。由此得到各节点的状态，最后模拟出系统的状态。

* 若系统正常，则将寿命增加
* 若系统异常，或总时间大于，输出寿命

此方法可以得出正确结果，且耗时较短。在20个节点的情况下，一次计算耗时约。

因此，本文采用变步长算法，设定仿真颗粒度为。

## 算法合理性说明

2.2中的算法可以规避以下两问题：

* 系统失效又复活

当系统异常时，算法会直接跳出循环，时间不会继续增加，即系统不会“复活”。

* 系统永生不死

当寿命大于时，算法自动跳出循环，输出，故系统不存在永生不死的问题。

因此，本文的算法具有合理性。

# 最大可靠性

对5~20个节点进行仿真，得到结果如下：

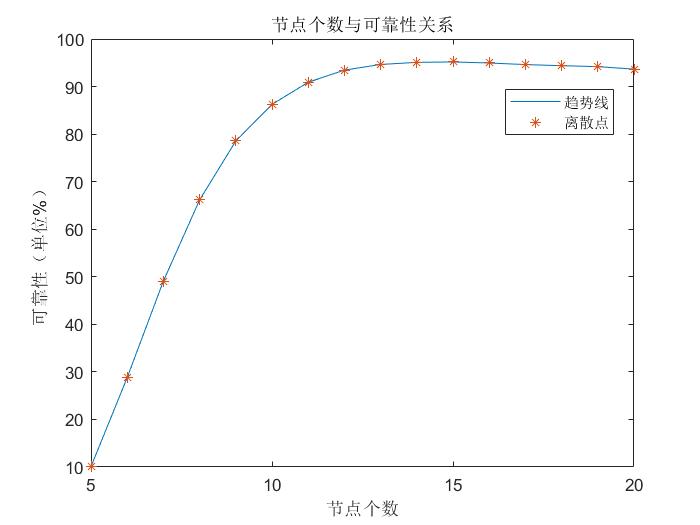


图 5最大可靠性

可以看出，节点个数较少时，系统容易发生“工作节点不足”的情况，导致寿命小于；节点个数较多时，总体故障概率增大，更有可能在前发生“总线阻塞”情况。因此，可靠性在节点个数适中时最大。

节点个数：15

最大可靠性：

# 最大平均工作寿命

对5~20个节点进行仿真，得到结果如下：

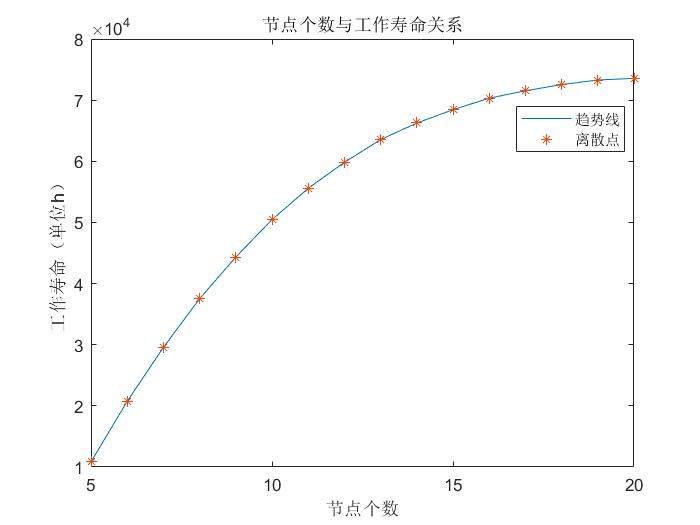


图 6最大平均工作寿命

可以看出，工作寿命随节点个数增多而增大，增速渐渐放缓，一方面是因为最大寿命的限制，另一方面则是节点个数增多导致“总线阻塞”情况更容易发生。

节点个数：20

最大平均工作寿命：

# 拓展部分

## 概述

本文希望通过不同方法，提升系统性能，包括可靠性与最大寿命这两方面。

对此，本文提出了提升最大限定寿命，提升切换器性能这两种方案。

## 提升最大限定寿命

在进行了20节点情况的模拟后，本文发现有系统，其失效不是因为内部故障导致，而是因为寿命触及了最大值，强制失效。

因此，本文希望通过提升最大限定寿命，减少系统寿命触及最大值的概率，从而提升系统的平均寿命。在接下来的模拟中，保持节点个数为20，每次将最大限定寿命提升，直到系统寿命触及最大值的概率小于。

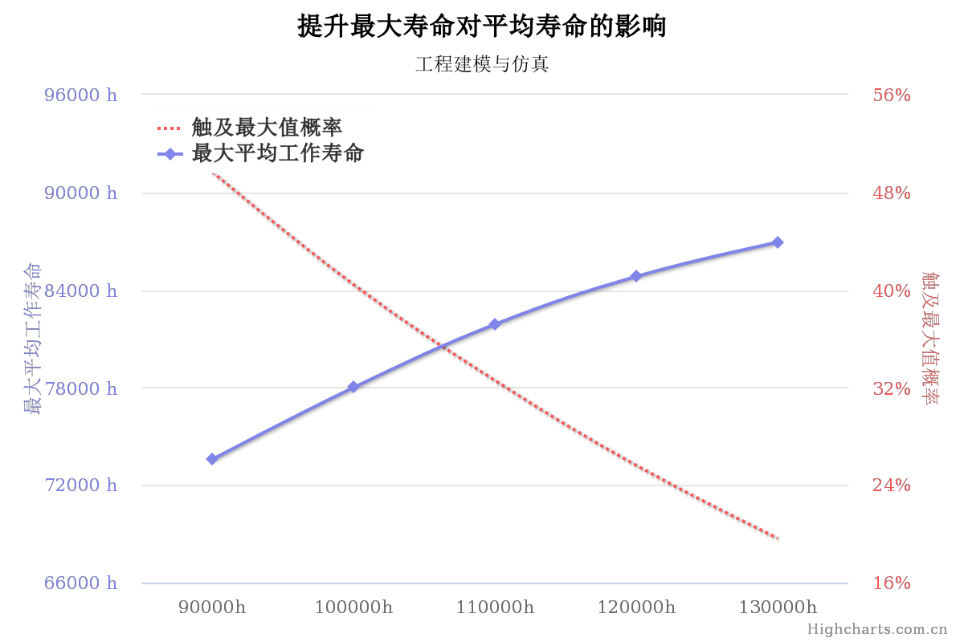


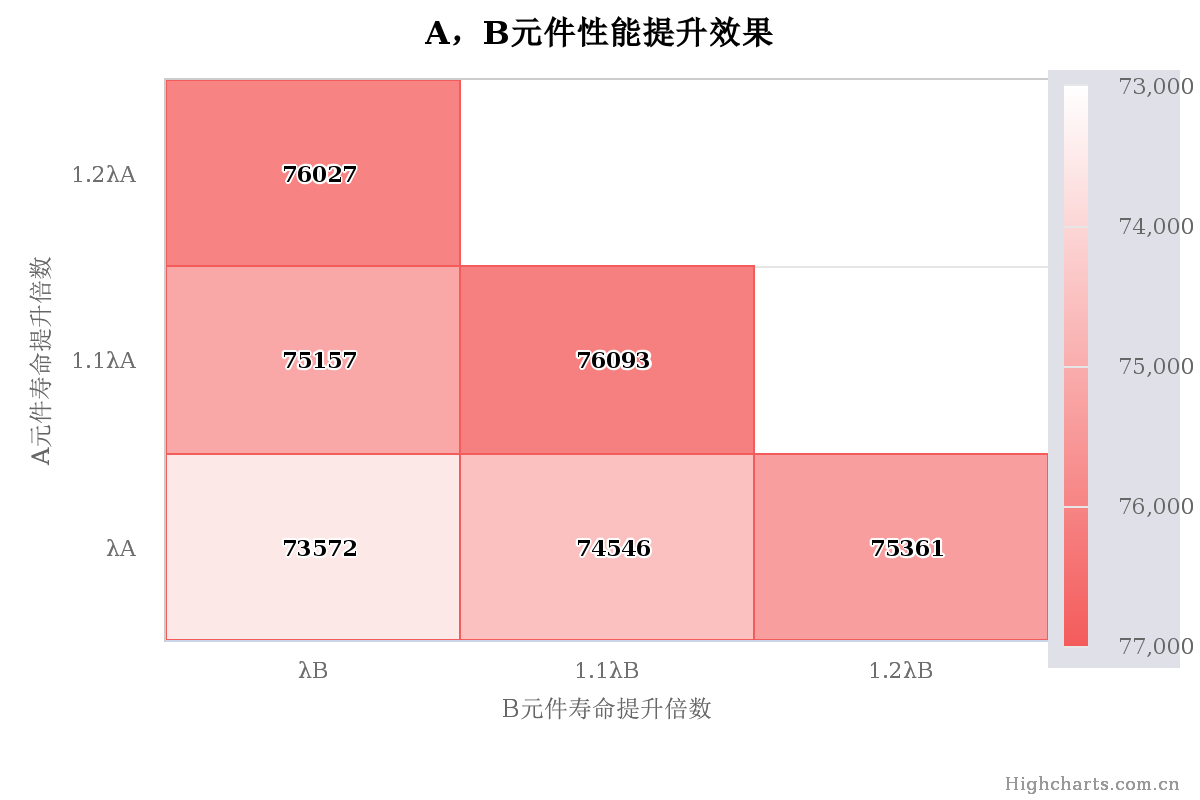
图 7提升最大寿命对平均寿命的影响

可以看出，随着最大限定寿命的增加，系统触及最大值的概率减小，因此系统的平均寿命增大。然而，当最大限定寿命足够大时，触及最大值的概率减小幅度不会很大，因此该方法在前期效果明显，后期（触及最大值概率为19.68%）对平均寿命的提升不足，需要探索新方法。

## 提升切换器性能

每个节点由切换器A，B构成，如果能减小切换器的故障概率，即可减小每一个节点的故障概率，从而增加整个系统的预期寿命。

切换器A，B发生故障的情况与有关。因此，将的值调整为原来的1.1，1.2倍，再调用函数，可以提升整体的预期寿命。以下热力图展示了这一方法的效果：



可以看出，提高可以提高单个切换器的性能，从而提升整体性能，其中提升A切换器性能的效果更为出色。同时，实验结果中的说明，与其专注于提升单个切换器性能，不如同时关注两个切换器，将其性能分别进行适度的提升。

# 总结

## 当下的优点与不足

优点：

* 选择了变步长方案，并证明了其时间远远优于定步长方法
* 规避了算法中“复活”与“永生”的漏洞
* 得出了稳定性与平均寿命最大情况下的结点个数，并进行了分析
* 提出了两种方案优化系统寿命，并采用多种可视化方式展示了优化结果

不足：

* 拓展部分对提升寿命关注较大，暂无提升可靠性的方案
* 运行时间仍有优化空间，运行计时中显示函数占用了运行时间的一半

## 优化方向

* 拓展部分新策略

拓展部分提出的两种方案，分别是：

针对整个系统：提升最大限定寿命

针对A，B元件：提升切换器性能

由于系统由切换器—节点—系统三级构成，因此可以探索一种针对节点的优化方 法，如防止节点失效导致的“总线阻塞”等，具体内容需要与老师进一步探讨。

* 不同方案下的结点数说明

在时，平均寿命最大的节点数为20，但当， 变化，或最大限定寿命升高时，最优节点数不一定仍为20个，应在优化后，对节 点数为5~20的情况依次进行尝试。

**参考文献**

[1]上海交通大学 电子工程系 工程问题建模和仿真讲座案例 2 问题和基本数学方法.

**附录**

变步长代码

Nsample=100000;

k=5; %工作结点数

pA=[0.26,0.26,0.48];

pB=[0.35,0.65];

pointOutput=zeros(1,n);

resOutput=zeros(1,n);

for numOfPoint=5:20 %总结点数

life=zeros(1,Nsample);

for t=1:Nsample

stateSystem=0;

stateComp=zeros(1,numOfPoint);

stateItemA=zeros(1,numOfPoint);

stateItemB=zeros(1,numOfPoint);

while stateSystem==0

[stateItemA,stateItemB,T]=changeStateItemT(stateItemA,stateItemB,pA,pB,numOfPoint);

stateComp=changeStateComp(stateItemA,stateItemB);

stateSystem=judgeSystem(stateComp);

life(t)=life(t)+T;

if life(t)>=90000

life(t)=90000;

break

end

end

%if mod(t,100)==0

% fprintf('%d sample,meanLife=%f \n',t,sum(life)/t)

%end

end

pointOutput(numOfPoint)=mean(life);

resOutput(numOfPoint)=sum((life-25000)>=0)\*100/Nsample;

fprintf('numOfPoint=%d, meanLife=%f \n',numOfPoint,pointOutput(numOfPoint))

fprintf('numOfPoint=%d, meanRes=%f %% \n',numOfPoint,resOutput(numOfPoint))

end

function [stateItemA,stateItemB,T]=changeStateItemT(stateItemA,stateItemB,pA,pB,n)

% 改状态

lambdaA=37000;

%lambdaA=lambdaA\*1.1;

lambdaB=480000;

%lambdaB=lambdaB\*1.1;

T1=exprnd(lambdaA,n,1);

T2=exprnd(lambdaB,n,1);

T1min=min(T1);

T2min=min(T2);

T1Index=find(T1==T1min);

T2Index=find(T2==T2min);

if T1min<=T2min && stateItemA(T1Index)==0

path=rand(1);

if path<pA(1)

stateItemA(T1Index)=1;

elseif pA(1)<=path<(pA(1)+pA(2))

stateItemA(T1Index)=2;

else

stateItemA(T1Index)=3;

end

end

if T1min>=T2min && stateItemB(T2Index)==0

path=rand(1);

if path<pB(1)

stateItemB(T2Index)=1;

else

stateItemB(T2Index)=2;

end

end

T=min(T1min,T2min);

end

function stateComp=changeStateComp(stateItemA,stateItemB)

%AB工作状态决定整体

changeMat=[0 3 1;1 5 1;2 3 4;4 4 4];

p=length(changeMat);

stateComp=changeMat(stateItemA+stateItemB\*p+1);

end