

# 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
1.1.	引言 .....	1
1.2.	寿命物理模型 .....	1
1.2.1.	切换器对节点 .....	1
1.2.2.	节点对系统 .....	2
1.3.	优化目标 .....	2
2.	算法选择 .....	3
2.1.	定步长 .....	3
2.2.	变步长 .....	3
2.3.	算法合理性说明 .....	4
3.	最大可靠性 .....	4
4.	最大平均工作寿命 .....	5
5.	拓展部分 .....	5
5.1.	概述 .....	5
5.2.	元件角度：提升切换器性能 .....	5
5.3.	节点角度：改变节点内部构造 .....	6
5.4.	系统角度：提升最大限定寿命 .....	7
6.	总结 .....	8
6.1.	当下的优缺点、优化方向 .....	8
6.2.	结论 .....	8

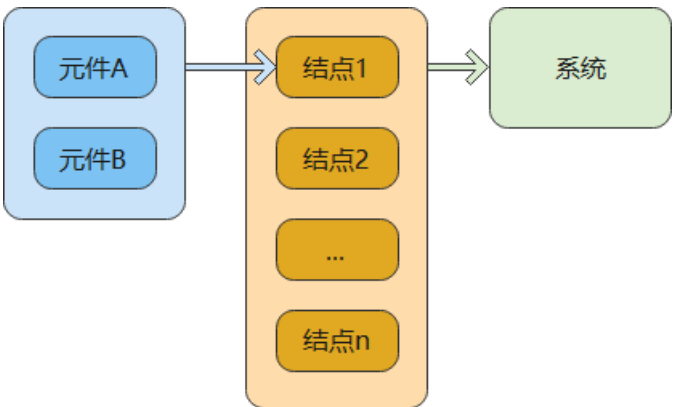
# 1 . 概述

## 1.1.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



## 1.2.寿命物理模型

### 1.2.1. 切换器对节点

切换器 A 状态	切换器 B 状态	节点状态	别名
$g_{A0}$	$g_{B0}$	$g_{N0}$	$g_{PF}$
	$g_{B1}$	$g_{N3}$	$g_{MO}$
	$g_{B2}$	$g_{N1}$	$g_{SO}$
$g_{A1}$	$g_{B0}$	$g_{N1}$	$g_{SO}$
	$g_{B1}$	$g_{N5}$	$g_{FB}$
	$g_{B2}$	$g_{N1}$	$g_{SO}$
$g_{A2}$	$g_{B0}$	$g_{N2}$	$g_{DM}$
	$g_{B1}$	$g_{N3}$	$g_{MO}$
	$g_{B2}$	$g_{N4}$	$g_{DN}$
$g_{A3}$	$g_{B0}$	$g_{N4}$	$g_{DN}$
	$g_{B1}$	$g_{N4}$	$g_{DN}$
	$g_{B2}$	$g_{N4}$	$g_{DN}$

表 1 节点状态对应表[1]

以上为切换器与节点状态的对应表，切换器 A, B 分别有 3, 2 种异常状态，这  $(3 + 1) \times (2 + 1) = 12$  种工作状态的组合，会产生 6 种不同的节点状态，这些状态的具体情况如下：

- $g_{N0}$ ：节点性能完好
- $g_{N1}$ ：只能作为从节点
- $g_{N2}$ ：或者作为主节点，或者作为不阻塞总线的失效节点
- $g_{N3}$ ：只能作为主节点，否则就会阻塞总线
- $g_{N4}$ ：成为不阻塞总线的失效节点
- $g_{N5}$ ：节点总是阻塞总线

### 1.2.2. 节点对系统

本题中，只有  $k = 5$  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。

基于 1.2.1 中 6 种不同的节点状态，可以将  $t$  时刻六种节点的个数记为  $Q_{N0}(t) \sim Q_{N5}(t)$ ，可知节点与系统工作状态的关系。[2]

系统在以下任意情况下不能工作：

- C1：任一节点处于  $g_{N5}$ ，即  $Q_{N5}(t) \geq 1$
- C2：有两个或以上节点处于  $g_{N3}$ ，即  $Q_{N3}(t) \geq 2$
- C3：无法找到适合工作于主模式的节点，即  $Q_{N0}(t) + Q_{N2}(t) + Q_{N3}(t) = 0$
- C4：能构成有效主从系统的节点总数少于  $k$  个， $Q_{N0}(t) + Q_{N1}(t) + u(Q_{N2}(t) + Q_{N3}(t)) < k$

系统满足在  $C5 \cap (C6 \cup C7)$  时能工作：

- C5：无节点处于  $g_{N5}$ ，即  $Q_{N5}(t) = 0$
- C6：有且仅有一个节点处于  $g_{N3}$ ，即  $Q_{N3}(t) = 1$  且  $Q_{N0}(t) + Q_{N1}(t) \geq k - 1$
- C7：无节点处于  $g_{N3}$ ，至少一个节点处于  $g_{N0}$  且该节点担当主节点，其余可工作于从模式的节点数不少于  $k - 1$ ，即  $Q_{N3}(t) = 0$  的前提下满足  $Q_{N0}(t) \geq 1$  且  $Q_{N0}(t) + Q_{N1}(t) \geq k - 1$ ，或满足  $Q_{N0}(t) = 0$  且  $Q_{N2}(t) \geq 1$  且  $Q_{N1}(t) \geq k - 1$

系统在同时满足以下情况时有概率能工作：

- C8：无节点处于  $g_{N5}$ ，即  $Q_{N3}(t) + Q_{N5}(t) = 0$
- C9：处于  $g_{N0}$  的节点大于等于 1，且处于  $g_{N0}$  和  $g_{N1}$  的节点总数恰好为  $k - 1$ ，另有若干节点处于  $g_{N2}$ ，即  $Q_{N0}(t) \geq 1$  且  $Q_{N0}(t) + Q_{N1}(t) = k - 1$  且  $Q_{N2}(t) \geq 1$

由于主节点随机选取， $p(\text{正常工作}) = \frac{Q_{N2}(t)}{Q_{N0}(t) + Q_{N2}(t)}$

### 1.3. 优化目标

- 系统可靠性  
若系统在  $[0, 25000h]$  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  $25000h$ ，即进行多次模拟，使得寿命超过  $25000h$  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。

## 2. 算法选择

### 2.1. 定步长

按照以下步骤设计算法：

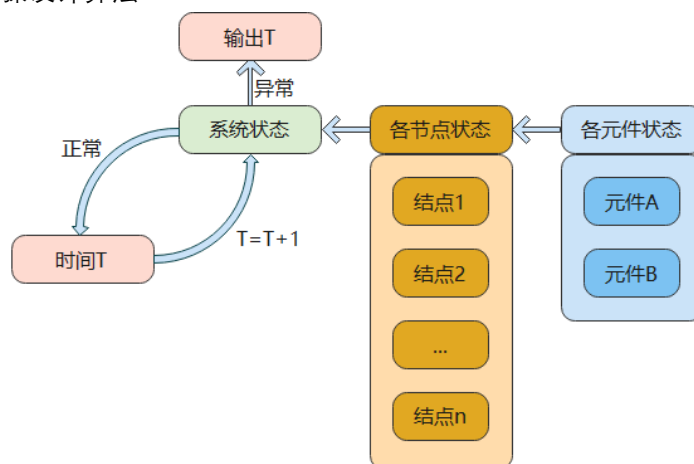


图 2 定步长算法示意图

设定步长为 $1h$ ，每步改变各元件的状态，改变概率 $p = 1 - e^{-\lambda}$ ，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 $1h$
- 若系统异常，或总时间大于 $90000h$ ，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 $133s$ 。由于使用数值模拟方法，样本数至少为 $10^5$ ，因此一次计算会耗时接近 $37h$ ，因此需要探索速度更快的方法。

### 2.2. 变步长

按照以下步骤设计算法：

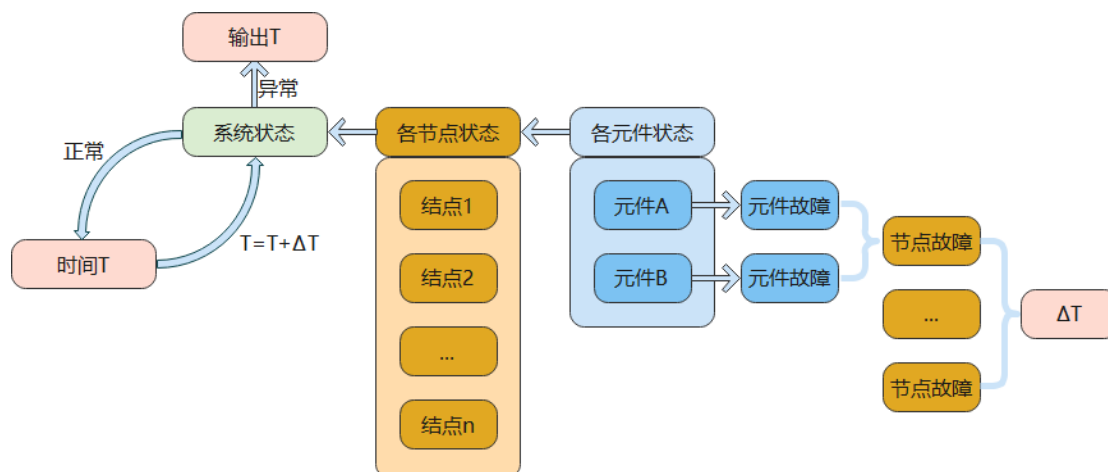


图 3 变步长算法示意图

在 $n$ 个节点中，依次模拟 $2n$ 个切换器 $A, B$ 的状态，得到各个切换器发生故障的时间 $T$ ， $f(T) = \lambda e^{-\lambda T}$ ，将最短的时间记为 $\Delta T$ ，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 $\Delta T$
- 若系统异常，或总时间大于 $90000h$ ，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300s。

因此，本文采用变步长算法，设定仿真颗粒度为 $10^5$ 。

## 2.3. 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 $T$ 不会继续增加，即系统不会“复活”。
- 系统永生不死  
当寿命大于 $90000h$ 时，算法自动跳出循环，输出 $T = 90000h$ ，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。

## 3. 最大可靠性

对5~20个节点进行仿真，得到结果如下：

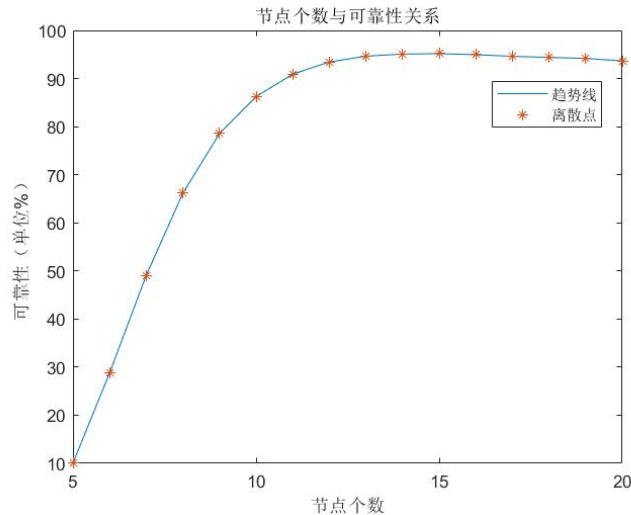


图 4 最大可靠性图

节点个数	13	14	15	16	17
可靠性	94.649%	95.086%	95.176%	94.959%	94.620%

表 2 最大可靠性图

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 $25000h$ ；节点个数较多时，总体故障概率增大，更有可能在 $25000h$ 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：95.176%

## 4. 最大平均工作寿命

对5~20个节点进行仿真，得到结果如下：

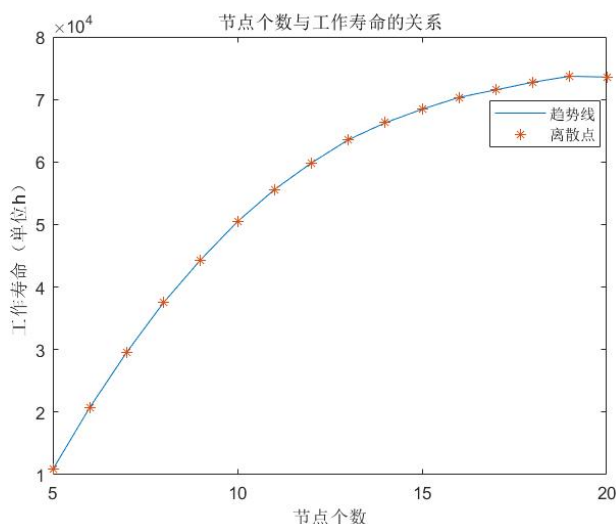


图 5 最大工作寿命图

节点个数	16	17	18	19	20
寿命 (h)	70326	71531	72749	73707	73572

表 3 最大工作寿命表

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90000h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：19

最大平均工作寿命：73707h

## 5. 拓展部分

### 5.1. 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，本文根据元件、节点、系统的“三步走”策略，提出了提升切换器性能，改变节点内部构造，增加最大限定寿命这三种方案。

### 5.2. 元件角度：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 $\lambda_A$ ,  $\lambda_B$ 有关。因此，将 $\lambda_A$ ,  $\lambda_B$ 的值调整为原来的1.1, 1.2倍，再调用 $exprnd(\lambda)$ 函数，可以提升整体的预期寿命与可靠性。在此，默认节点个数为20，以平均寿命的提升为例，以下热力图展示了这一方法的效果：

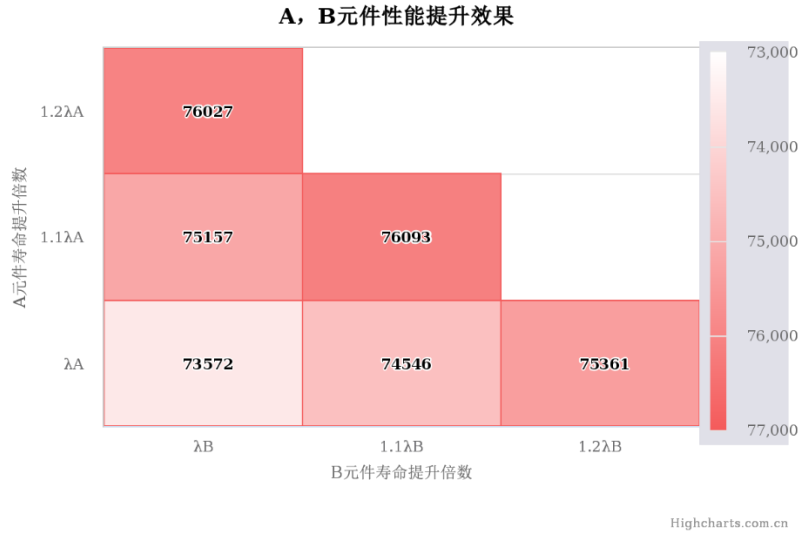


图 6 切换器寿命与总体寿命的关系

可以看出，提高 $\lambda_A$ ,  $\lambda_B$ 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(76093 > 76027)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。

### 5.3. 节点角度：改变节点内部构造

初始的节点设计中，包含一个A切换器和一个B切换器，根据[3]中的分析可知，“总线阻塞”对系统寿命影响大，它主要跟切换器B故障有关。因此，额外设置一个切换器B，通过两个B串联的方式，可以有效地减少总线阻塞。

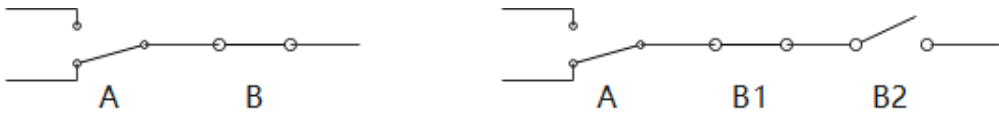


图 7 传统节点构造与新节点构造

可以看出，当传统节点的B掷刀无法与触点脱离时，若A也存在故障，容易发生总线阻塞。然而，新接线方式中，即使 $B_1$ 处于常闭合状态， $B_2$ 的工作状态不受任何影响，因此发生总线阻塞的概率大大降低。

将 $B_1$ 与 $B_2$ 视为一个统一的元件，其工作状态的组合会决定整体的状态：

$B_1$	$B_2$	正常工作	常闭合状态	常断开状态
正常工作		正常工作	正常工作	常断开状态
常闭合状态		正常工作	常闭合状态	常断开状态
常断开状态		常断开状态	常断开状态	常断开状态

表 4 整体工作状态与 $B_1$ ,  $B_2$ 的关系

同时使用新旧节点构造，对比18，19，20个节点状态下的工作寿命与可靠性：

节点数	旧寿命(h)	新寿命(h)	提升比例	旧可靠性	新可靠性	提升比例
18	72749	81413	11.914%	94.386%	99.949%	5.894%
19	73707	82798	12.333%	94.192%	99.985%	6.150%
20	73572	83998	14.171%	93.665%	99.988%	6.751%

表 5 新旧节点构造效果对比

可以看出，改变节点内部构造的作用在于“防止系统寿命过短”，即防止总线阻塞的情况发生。因此，系统寿命有了大幅度的提升。值得一提的是，由于总线阻塞的情况发生概率大大减少，系统的寿命通常大于25000h，因此提升可靠性效果显著，可以获得接近100%的可靠性。

## 5.4.系统角度：提升最大限定寿命

在进行了20节点情况的模拟后，本文发现有系统近一半，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90000h，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10000h，直到系统寿命触及最大值的概率小于20%。

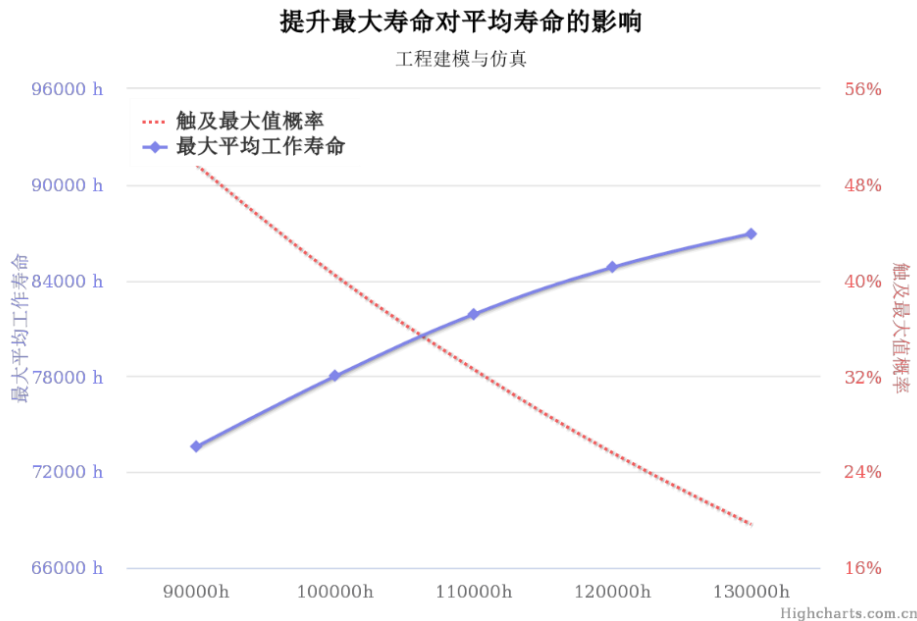


图 8 提升最大寿命对平均寿命的影响

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，后期对平均寿命的提升不足。因此，需要了解整体的寿命分布情况，才可以选择合适的最大限定寿命，尽量减少强制失效。

依然以节点个数20为例，以下为系统寿命的分布图(在原始设置下，系统寿命73572h)，可以看出，系统的寿命曲线类似于高斯分布。因此，采用取点法，分别强制丢弃20%，10%，5%的系统，观察寿命提升效果：



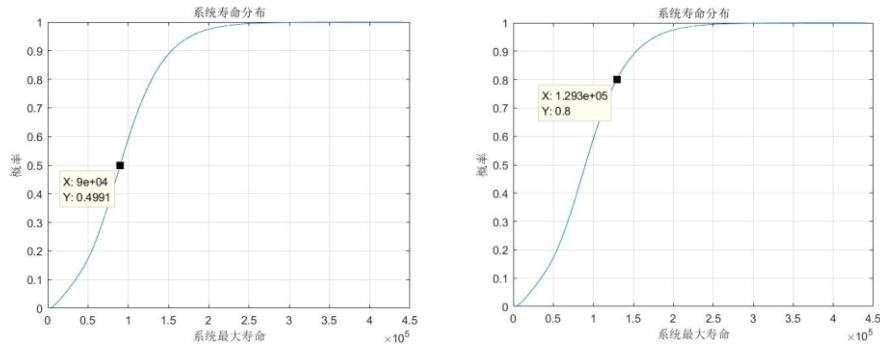


图 9 系统寿命分布

强制失效	最大限定寿命 ( $h$ )	系统平均寿命 ( $h$ )	提升比例
20%	129300	86982	11.823%
10%	153900	90393	12.286%
5%	177500	91787	12.476%

表 6 不同丢弃比例对平均寿命的影响

结论显示，最大限定寿命越大，强制失效比例越小，系统的平均寿命越大，符合预期。

## 6. 总结

### 6.1. 当下的优缺点、优化方向

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
- 提出元件-节点-系统三级方案优化系统寿命，并采用多种可视化方式展示优化结果

不足：

- 系统角度的优化方案只能提升寿命，无法像元件、节点角度方案一样提升可靠性
- 运行时间仍有优化空间，运行计时中显示`exprnd`函数占用了运行时间的一半
- 不同方案下的结点数说明

优化方向：

- 当 $\lambda_A$ ,  $\lambda_B$ 变化，或最大限定寿命升高时，最优节点数不一定为20个，应在三种优化方案中，对节点数为5~20的情况依次尝试。（因算力问题只以20节点为例）

### 6.2. 结论

通过本文研究发现，在题目条件下，节点数为15时，可靠性95.176%最大；节点数为19时，寿命73707h最大。

从元件维度，可以通过提升 $A$ ,  $B$ 各自的性能来提高整体寿命与可靠性，效果较好；

从节点维度，可以增加一个 $B$ 元件提升寿命与可靠性，其中可靠性增至近100%；

从系统维度，可以提升最大限定寿命从而大幅提升平均寿命，但对可靠性无作用。

## 附录

### 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 15
- [2]上海交通大学 电子工程系 案例 2 系统可用性数值的理论求解方法介绍
- [3]上海交通大学 电子工程系 工程问题建模与仿真讲座 16

### 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);
        stateltemA=zeros(1,numOfPoint);
        stateltemB=zeros(1,numOfPoint);
        while stateSystem==0

[stateltemA,stateltemB,T]=changeStateltemT(stateltemA,stateltemB,pA,pB,numOfPoint);
            stateComp=changeStateComp(stateltemA,stateltemB);
            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 [stateltemA,stateltemB,T]=changeStateltemT(stateltemA,stateltemB,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 && stateltemA(T1Index)==0
    path=rand(1);
    if path<pA(1)
        stateltemA(T1Index)=1;
    elseif pA(1)<=path<(pA(1)+pA(2))
        stateltemA(T1Index)=2;
    else
        stateltemA(T1Index)=3;
    end
end

if T1min>=T2min && stateltemB(T2Index)==0
    path=rand(1);
    if path<pB(1)
        stateltemB(T2Index)=1;
    else
        stateltemB(T2Index)=2;
    end
end
T=min(T1min,T2min);
end

function stateComp=changeStateComp(stateltemA,stateltemB)
%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(stateltemA+stateltemB*p+1);
end

```