**Лабораторная работа №4**

“Статистическое моделирование функционирования системы с запасными элементами”

Вариант № 3

|  |  |
| --- | --- |
| Выполнил  студент гр. 33504/2: | Лелюхин Д. О. |
| Проверил: | Чуркин В.В. |

# Задание



**Рисунок 1. Структурно-надежностная схема системы.**

*m* = 3;  1/ч; *P*0 = 0,99; *T* = 8760 ч

# Ход работы

Возьмем:

ЛФРС системы:

Код программы на языке R:

LFSR=function(x)

{

T <- 8760

return ((((x[1]>T)&(x[2]>T))|(x[3]>T))&(x[4]>T)&(x[5]>T)&((x[6]>T)|(x[7]>T)|(x[8]>T)))

}

trace = function(L)

{

N<-53562

n<-c(3,2,3)

m<-3

lambda<-c(40\*10^(-6), 10\*10^(-6), 80\*10^(-6))

d<-0

for(k in 1:N)

{

x <-c()

for(i in 1:m)

{

t <-c()

for(j in 1:n[i])

{

alfa<-runif(1,min=0, max=1)

t <- c(t,(-log(alfa)/lambda[i]))

}

for(j in 1:L[i])

{

l<-which.min(t)

t[l] <- t[l]-log(runif(1,min=0, max=1))/lambda[i]

}

for(j in 1:n[i])

{

x <- c(x,t[j])

}

}

d <- d + !LFSR(x)

}

P <- 1-d/N

return (P)

}

P0 <- 0.99

L<-integer(3)

for (i in 1:4){

L[1]<-i

for(j in 1:4){

L[2]<-j

for(k in 1:4){

L[3]<-k

P <- trace(L)

if(P>P0){

print('P=')

print(P)

print(L)

print(sum(L))}

}

}

}

Код программы на языке С:

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <math.h>

#include <string.h>

#include <time.h>

double min(double \*t)

{

double min = t[0];

for (int i = 0; i < sizeof(t)/sizeof(t[0]); ++i)

{

if (min > t[i])

{

min = t[i];

}

}

return min;

}

bool LFRS(int \*x, int T)

{

return (((x[0]>T)&(x[1]>T))|((x[2]>T)&(x[3]>T)))&(x[4]>T)&(x[5]>T)&(((x[6]>T)&(x[7]>T))|((x[8]>T)&(x[9]>T))|((x[10]>T)&(x[11]>T)));

}

double trace(int \*L, long int N, unsigned int T, int m, double \*lam, int \*n)

{

srand(time(NULL));

unsigned int d;

double P;

//int \*x;

//x = (int \*) malloc(13\*sizeof(int));

int x[12];

double \*t;

t = (double \*) malloc(sizeof(double)\*(T^2));

double alpha;

unsigned long int l;

d = 0;

for (int k = 0; k < N; ++k)

{

memset(x, 0, sizeof(x));

for (int i = 0; i < m; ++i)

{

memset(t, 0, sizeof(t));

for (int j = 0; j < sizeof(n)/sizeof(n[0]); ++j)

{

alpha = (double)rand() / (double)RAND\_MAX;

t = (double\*) realloc(t, sizeof(t)\*sizeof(double) + sizeof(double));

t[sizeof(t)] = -log(alpha)/lam[i];

}

for (int j = 0; j < sizeof(L)/sizeof(L[0]); ++j)

{

l = (int) min(t);

if (l > sizeof(t))

{

t = (double\*) realloc(t, sizeof(double)\*l + sizeof(double));

}

t[l] = t[l]-log(rand() % 2)/lam[i];

}

for (int j = 0; j < sizeof(n)/sizeof(n[0]); ++j)

{

x = (int \*) realloc(x, sizeof(x)\*sizeof(int) + sizeof(int));

x[sizeof(x)] = t[j];

}

}

d = d + !LFRS(x, T);

}

free(t);

P = 1 - (double) d/N;

return P;

}

int main()

{

clock\_t begin = clock();

FILE \*pFile;

pFile = fopen("out.txt","w");

long int N;

unsigned int sum;

const unsigned int m = 3;

const unsigned int T = 8760;

const double P0 = 0.999;

double P;

const double eps = 0.001;

const double ta = 3.715;

double lam[3] = {40e-06, 10e-06, 80e-06};

int n[3] = {4, 2, 6};

int L[3];

N = pow(ta,2)\*(P0\*(1-P0))/pow(eps,2);

for (int i = 0; i < 4; ++i)

{

L[0] = i + 1;

for (int j = 0; j < 4; ++j)

{

L[1] = j + 1;

for (int k = 0; k < 4; ++k)

{

L[2] = k + 1;

P = trace(L, N, T, m, lam, n);

//if (P > P0)

//{

sum = 0;

fprintf(pFile, "\tP= %f\n", P);

printf("\tP= %f\n", P);

fprintf(pFile, "\tL= ");

printf("\tL= ");

for (int v = 0; v < 4; ++v)

{

fprintf(pFile, " %i ", L[v]);

printf(" %i ", L[v]);

sum = sum + L[v];

}

fprintf(pFile, "\n\tSum(L)=%i\n\n", sum);

printf("\n\tSum(L)=%i\n\n", sum);

//}

printf("Still works ... %i %i %i %i\n", i, j, k);

}

}

}

clock\_t end = clock();

double time\_spent = (double)(end - begin) / CLOCKS\_PER\_SEC;

printf("%f\n", time\_spent);

return 0;

}

Результаты:

[1] "P="

[1] 0.990497

[1] 3 2 3

[1] 8

[1] "P="

[1] 0.9957059

[1] 3 2 4

[1] 9

[1] "P="

[1] 0.9920466

[1] 3 3 3

[1] 9

[1] "P="

[1] 0.9964527

[1] 3 3 4

[1] 10

[1] "P="

[1] 0.9916545

[1] 3 4 3

[1] 10

[1] "P="

[1] 0.996042

[1] 3 4 4

[1] 11

[1] "P="

[1] 0.9924573

[1] 4 2 3

[1] 9

[1] "P="

[1] 0.9970875

[1] 4 2 4

[1] 10

[1] "P="

[1] 0.9930361

[1] 4 3 3

[1] 10

[1] "P="

[1] 0.997909

[1] 4 3 4

[1] 11

[1] "P="

[1] 0.9933722

[1] 4 4 3

[1] 11

[1] "P="

[1] 0.997965

[1] 4 4 4

[1] 12

Оптимальный результат:

[1] "P="

[1] 0.990497

[1] 3 2 3

[1] 8

# Вывод

Оптимальное количество запасных элементов(L1,L2,L3): (3,2,3), тогда вероятность безотказной работы будет равна 0,99 за 8760 ч.