МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №4
по дисциплине «Качество и метрология программного обеспечения»
Тема: Построение операционной графовой модели программы (ОГМП) и расчет характеристик эффективности ее выполнения методом эквивалентных преобразований

Студент гр. 8304 Алтухов А.Д.

Преподаватель Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Построение операционной графовой модели программы (ОГМП) и расчет характеристик эффективности ее выполнения методом эквивалентных преобразований.

Ход выполнения.

Для программы из первой лабораторной работы (приложение А) был построен граф управления программой, представленный на рисунке 1.

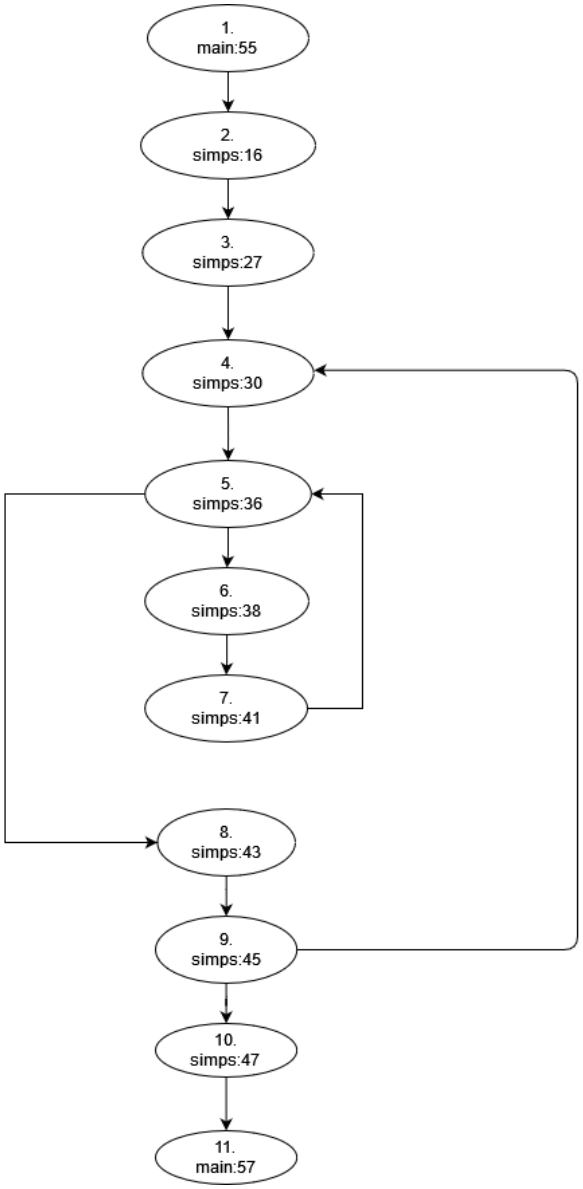

Рисунок 1 – Операционно-графовая модель

Проведен расчет вероятностей, приведенный в таблице 1.

Таблица 1. Расчет вероятностей

исх	**прием**	**общее время**	**кол-во проходов**	**среднее время**	**вероятность**
55	16	17.222	1	17.222	1
16	27	6872.222	1	6872.222	1
27	30	9.444	1	9.444	1
30	36	16.667	1	16.667	1
36	38	26.667	1	26.667	1
38	41	76.111	2	38.056	1
41	38	31.111	1	31.111	0.5
41	43	21.667	1	21.667	0.5
43	45	19.444	1	19.444	1
45	47	13.333	1	13.333	1
47	57	37.222	1	37.222	1

Операционная графовая модель с нагруженными дугами представлена на рисунке 2.

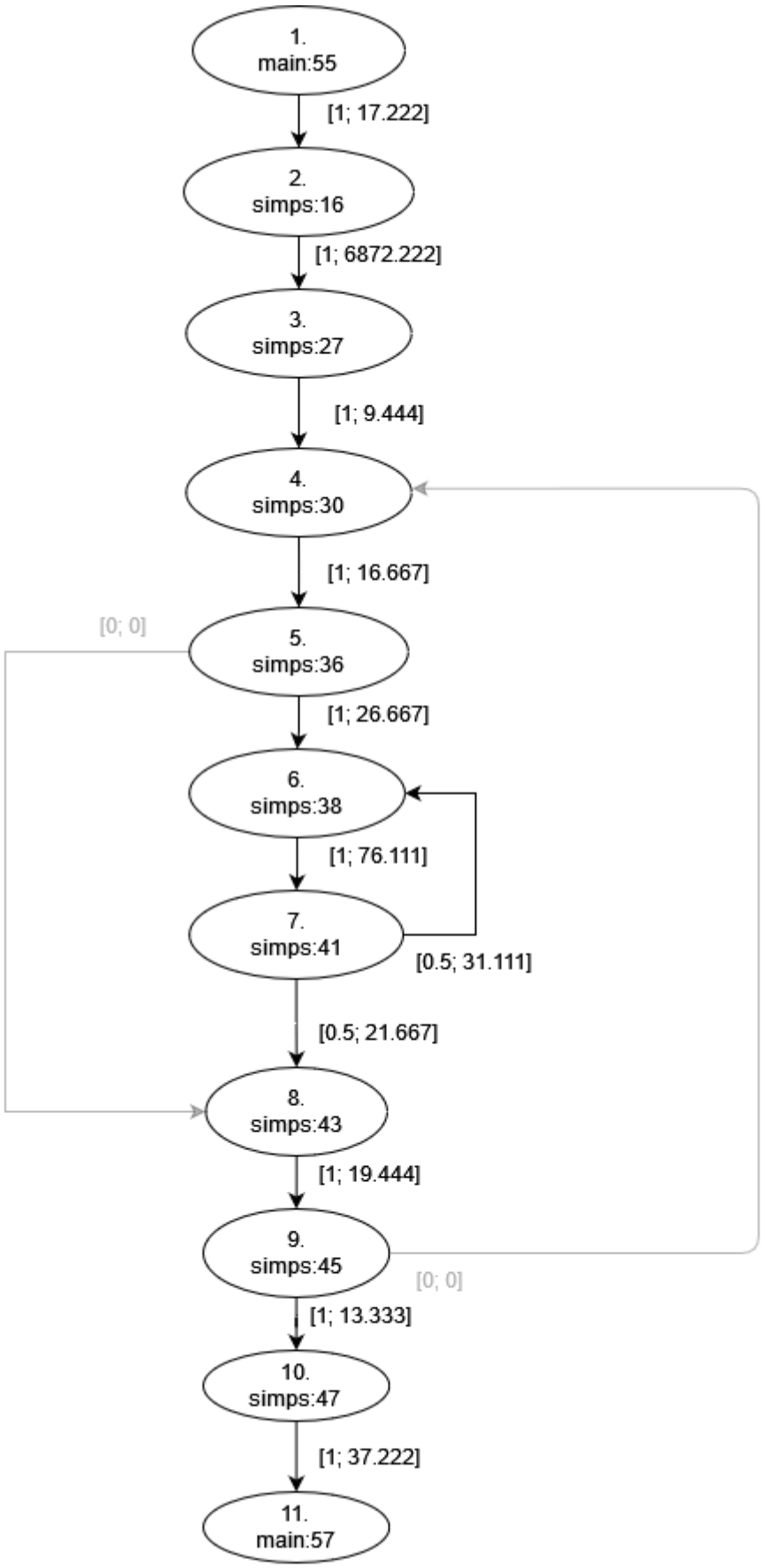

Рисунок 2 – Операционная графовая модель программы с нагруженными дугами

С помощью программного средства CSA III (описание графа представлено в приложении Б) выполнены эквивалентные преобразования над операционно-графовой моделью. Результат представлен в таблице 2. Графическое представление модели представлено на рисунке 3.

Таблица 2. Результат работы программы

probability	1.0
intensity	7217.221
deviation	22993.113568

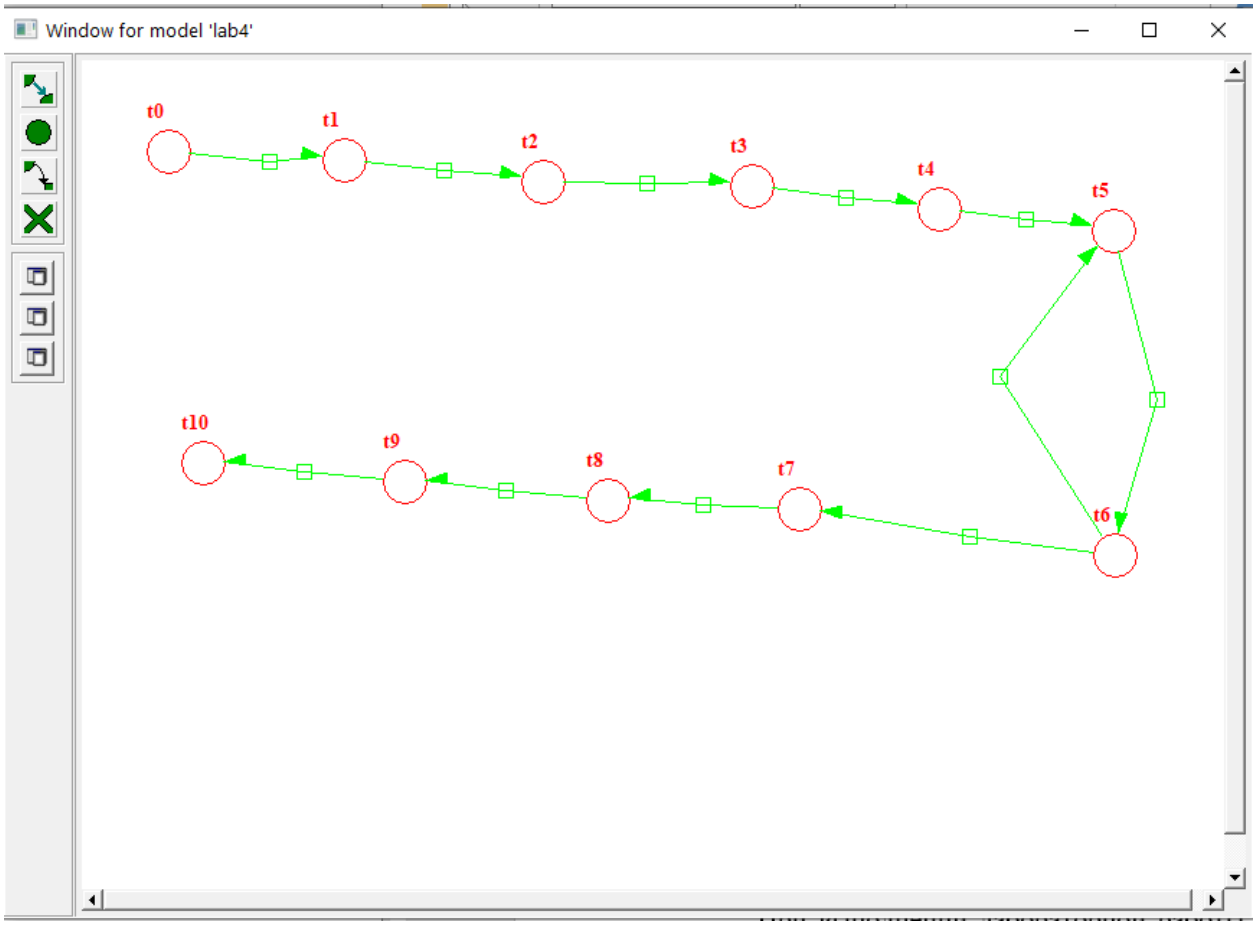

Рисунок 3 – Графовая модель программы

Выводы.

При выполнении лабораторной работы была построена операционная графовая модель заданной программы, нагрузочные параметры которой были оценены с помощью профилировщика Sampler и методом эквивалентных преобразований с помощью пакета CSA III были вычислены математическое ожидание и дисперсия времени выполнения как для всей программы, так и для заданного фрагмента. С помощью пакета CSA III были получены следующие результаты: дисперсия – 22993.113568 и математическое ожидание – 7217.221, что соответствует результатам, полученным с помощью Sampler – 7103.055.

Приложение А.

```
#include <stdio.h>
#include <math.h>
#include "sampler.h"

const double tol = 1.0E-6;

double fx(double x){
   return exp(-x/2);
}

double dfx(double x){
   return -(exp(-x/2))/2;
}

double simps(double lower, double upper, double tol, double* sum){
        SAMPLE;
   int pieces=2;
   double delta_x=(upper-lower)/pieces;
   double odd_sum = fx(lower+delta_x);
   double even_sum =0.0;
   double end_sum =fx(lower)+fx(upper);
   double end_cor =dfx(lower)-dfx(upper);
   *sum=(end_sum+4.0*odd_sum)*delta_x/3.0;

   double sum1;
   double x;
   SAMPLE;
   do
   {
```

```
                SAMPLE;
            pieces=pieces*2;
            sum1=*sum;
            delta_x=(upper-lower)/pieces;
            even_sum=even_sum+odd_sum;
            odd_sum=0.0;
                SAMPLE;
            for (int i=1; i<=pieces/2; i++) {
                    SAMPLE;
                x=lower+delta_x*(2.0*i-1.0);
                odd_sum=odd_sum+fx(x);
                    SAMPLE;
            }
                SAMPLE;

*sum=(7.0*end_sum+14.0*even_sum+16.00*odd_sum+end_cor*delta_x)*delta_x
/15.0;
                    SAMPLE;
          } while ( (*sum!=sum1) && (fabs(*sum-sum1)<=fabs(tol*(*sum))) );
              SAMPLE;
        }
        int main(int argc, char** argv)
        {
              sampler_init(&argc, argv);
           double lower=1.0;
           double upper=9.0;
           double sum = 0.0;
              SAMPLE;
           simps(lower,upper,tol,&sum);
              SAMPLE;
```

```
    //printf("\narea= %f\n", sum);
    return 0;
}
```

Приложение Б.

```
<model type = "Objects::AMC::Model" name = "lab4">
    <node type = "Objects::AMC::Top" name = "t0"></node>
    <node type = "Objects::AMC::Top" name = "t1"></node>
    <node type = "Objects::AMC::Top" name = "t2"></node>
    <node type = "Objects::AMC::Top" name = "t3"></node>
    <node type = "Objects::AMC::Top" name = "t4"></node>
    <node type = "Objects::AMC::Top" name = "t5"></node>
    <node type = "Objects::AMC::Top" name = "t6"></node>
    <node type = "Objects::AMC::Top" name = "t7"></node>
    <node type = "Objects::AMC::Top" name = "t8"></node>
    <node type = "Objects::AMC::Top" name = "t9"></node>
    <node type = "Objects::AMC::Top" name = "t10"></node>
    <link type = "Objects::AMC::Link" name = "t0-->t1" probability = "1.0" intensity = "17.222" deviation = "0.0" source = "t0" dest = "t1"></link>
    <link type = "Objects::AMC::Link" name = "t1-->t2" probability = "1.0" intensity = "6872.222" deviation = "0.0" source = "t1" dest = "t2"></link>
    <link type = "Objects::AMC::Link" name = "t2-->t3" probability = "1.0" intensity = "9.444" deviation = "0.0" source = "t2" dest = "t3"></link>
    <link type = "Objects::AMC::Link" name = "t3-->t4" probability = "1.0" intensity = "16.667" deviation = "0.0" source = "t3" dest = "t4"></link>
    <link type = "Objects::AMC::Link" name = "t4-->t5" probability = "1.0" intensity = "26.667" deviation = "0.0" source = "t4" dest = "t5"></link>
    <link type = "Objects::AMC::Link" name = "t5-->t6" probability = "1.0" intensity = "76.111" deviation = "0.0" source = "t5" dest = "t6"></link>
    <link type = "Objects::AMC::Link" name = "t6-->t7" probability = "0.5" intensity = "21.667" deviation = "0.0" source = "t6" dest = "t7"></link>
    <link type = "Objects::AMC::Link" name = "t8-->t9" probability = "1.0" intensity = "13.333" deviation = "0.0" source = "t8" dest = "t9"></link>
```

<link type = "Objects::AMC::Link" name = "t9-->t10" probability = "1.0" intensity = "37.222" deviation = "0.0" source = "t9" dest = "t10"></link>

<link type = "Objects::AMC::Link" name = "t6-->t5" probability = "0.5" intensity = "31.111" deviation = "0.0" source = "t6" dest = "t5"></link>

<link type = "Objects::AMC::Link" name = "t7-->t8" probability = "1.0" intensity = "19.444" deviation = "0.0" source = "t7" dest = "t8"></link>

</model>