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 _____ Бочаров Ф.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.

Задание

Построение ОГМП.

Для рассматривавшегося в лабораторных работах 1-3 индивидуального задания разработать операционную модель управляющего графа программы на основе схемы алгоритма. При выполнении работы рекомендуется для упрощения обработки графа исключить диалог при выполнении операций ввода-вывода данных, а также привести программу к структурированному виду.

Выбрать вариант графа с нагруженными дугами, каждая из которых должна представлять фрагмент программы, соответствующий линейному участку или ветвлению. При расчете вероятностей ветвлений, зависящих от распределения данных, принять равномерное распределение обрабатываемых данных в ограниченном диапазоне (например, [0,100] - для положительных чисел или [-100,100] - для произвольных чисел). В случае ветвлений, вызванных проверкой выхода из цикла, вероятности рассчитываются исходя априорных сведений о числе повторений цикла. Сложные случаи оценки вероятностей ветвлений согласовать с преподавателем.

В качестве параметров, характеризующих потребление ресурсов, использовать времена выполнения команд соответствующих участков программы. С помощью монитора Sampler выполнить оценку времен выполнения каждого линейного участка в графе программы.

Расчет характеристик эффективности выполнения программы методом эквивалентных преобразований.

Полученную в части 2.1 данной работы ОГМП, представить в виде графа с нагруженными дугами, у которого в качестве параметров, характеризующих потребление ресурсов на дуге ij, использовать тройку $\{P_{ij}, M_{ij}, D_{ij}\}$, где:

P_{ij} - вероятность выполнения процесса для дуги ij,

M_{ij} - мат. ожидание потребления ресурса процессом для дуги ij,

D_{ij} - дисперсия потребления ресурса процессом для дуги ij.

В качестве потребляемого ресурса в данной работе рассматривается время процессора, а оценками мат. ожиданий времен для дуг исходного графа следует принять времена выполнения операторов (команд), соответствующих этим дугам участков программы. Дисперсиям исходных дуг следует присвоить нулевые значения.

Получить описание полученной ОГМП на входном языке пакета CSA III в виде поглощающей марковской цепи (ПМЦ) – (англ.) AMC (absorbing Markov chain) и/или эргодической марковской цепи (ЭМЦ) – EMC (ergodic Markov chain).

С помощью предоставляемого пакетом CSA III меню действий выполнить расчет среднего времени и дисперсии времени выполнения как для всей программы, так и для ее фрагментов, согласованных с преподавателем. Сравнить полученные результаты с результатами измерений, полученными в работе 3.

Ход выполнения

Для программы из первой лабораторной работы (прил. А) была построена операционно-графовая модель (рис. 1) с помощью программы Sampler.

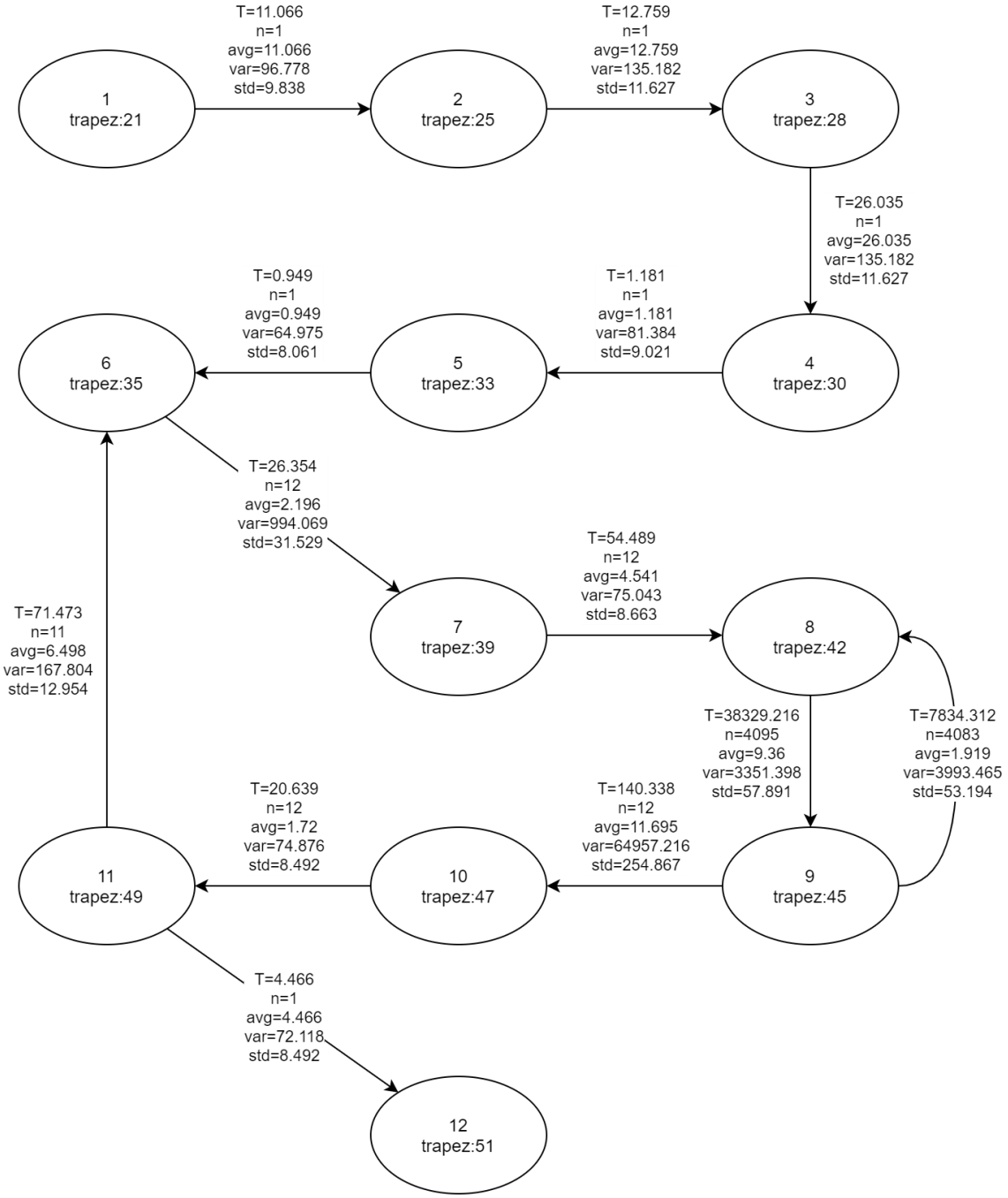

Рисунок 1. Операционно-графовая модель

Проведен расчет вероятностей (табл. 1):

исх	прием	общее время	кол-во проходов	вероятность	среднее время
21	25	11.066	1	1	11.066
25	28	12.759	1	1	12.759
28	30	26.035	1	1	26.035
30	33	1.181	1	1	1.181
33	35	0.949	1	1	0.949
35	39	26.354	12	1	2.196
39	42	54.489	12	1	4.541
42	45	38329.216	4095	1	9.360
45	47	140.338	12	0.00293	11.695
45	42	7834.312	4083	0.99707	1.919
47	49	20.639	12	1	1.720
49	35	71.473	11	0.91667	6.498
49	51	4.466	1	0.08333	4.466
		46533.277			

Таблица 1. Расчет вероятности для ОГМП

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

probability	1.00000000000005
intensity	46542.4463548265
deviation	2158738160.37573

Таблица 2. Результат работы CSA III

Вывод

В ходе выполнения лабораторной работы были изучены возможности построения операционной графовой модели программы (ОГМП) и расчёта характеристик эффективности её выполнения методом эквивалентных преобразований. С помощью пакета CSA III были получены следующие результаты: дисперсия – 2158738160.37573 и математическое ожидание – 46542.446, что соответствует результатам, полученным с помощью Sampler – 46533.277.

Приложение А

```
1.  #include "sampler.h"
2.  #include <stdio.h>
3.  #include <math.h>
4.
5.  /* integration by the trapezoidal rule */
6.
7.  const double tol = 1.0e-6;
8.  double sum, upper, lower;
9.
10. /* find f(x)=1/x */
11. /* watch out for x=0 ! */
12. double fx(double x) {
13.     return 1.0/x;
14. }
15.
16. /* numerical integration by the trapezoid method */
17. /* function is FX, limits are LOWER and UPPER */
18. /* with number of regions equal to PIECES */
19. /* fixed partition is DELTA_X, answer is SUM */
20. void trapez(double lower, double upper, double tol, double* sum) {
21.     SAMPLE;
22.     int pieces, i;
23.     double x, delta_x, end_sum, mid_sum, sum1;
24.
25.     SAMPLE;
26.     pieces = 1;
27.     delta_x = (upper - lower)/pieces;
28.     SAMPLE;
29.     end_sum = fx(lower) + fx(upper);
30.     SAMPLE;
31.     *sum = end_sum * delta_x / 2.0;
32.     mid_sum = 0.0;
33.     SAMPLE;
34.     do {
35.         SAMPLE;
36.         pieces = pieces*2;
37.         sum1 = *sum;
38.         delta_x = (upper-lower)/pieces;
39.         SAMPLE;
40.         for(i = 1; i <= pieces / 2; i++)
41.         {
42.             SAMPLE;
43.             x = lower + delta_x * (2.0 * i - 1.0);
44.             mid_sum = mid_sum + fx(x);
45.             SAMPLE;
46.         }
47.         SAMPLE;
48.         *sum = (end_sum + 2.0 * mid_sum) * delta_x * 0.5;
49.         SAMPLE;
50.     } while ( fabs(*sum - sum1) > fabs(tol*(*sum)));
51.     SAMPLE;
52. }
53.
54. int main(int argc, char **argv) {
55.     sampler_init(&argc, argv);
56.     lower = 1.0;
57.     upper = 9.0;
58.     trapez(lower,upper,tol,&sum);
59.     // printf("area= %.16e\n", sum);
60.     return 0;
61. }
62.
63.
```

Приложение Б

```
1.  Tops = { t1 , t2 , t3 , t4 , t5 , t6 , t7 , t8 , t9 , t10 , t11 , t12 , }
2.  Arcs = {
3.  [t1-t2] (21--25) { 1.0 , 11.066 },
4.  [t2-t3] (25--28) { 1.0 , 12.759 },
5.  [t3-t4] (28--30) { 1.0 , 26.035 },
6.  [t4-t5] (30--33) { 1.0 , 1.181 },
7.  [t5-t6] (33--35) { 1.0 , 0.949 },
8.  [t6-t7] (35--39) { 1.0 , 2.196 },
9.  [t7-t8] (39--42) { 1.0 , 4.541 },
10. [t8-t9] (42--45) { 1.0 , 9.360 },
11. [t9-t8] (45--42) { 0.99707 , 1.919 },
12. [t9-t10] (45--47) { 0.00293 , 11.695 },
13. [t10-t11](47--39) { 1.0 , 1.720 },
14. [t11-t6] (49--35) { 0.91667 , 6.498 },
15. [t11-t12](49--51) { 0.08333 , 4.466 },
16. }
17.
```