**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №5**

**по дисциплине «Качество и метрология программного обеспечения»**

**Тема: Оценка параметров надежности программ**

**по временным моделям обнаружения ошибок**

|  |  |  |
| --- | --- | --- |
| Студент гр. 7304 |  | Соколов И.Д. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2021

# **Формулировка задания**

Выполнить исследование показателей надежности программ, характеризуемых моделью обнаружения ошибок Джелинского-Моранды, для различных законов распределения времен обнаружения отказов и различного числа используемых для анализа данных. Для проведения исследования требуется:

1. Сгенерировать массивы данных {Xi}, где Xi – случайное значение *интервала между соседними (i-1)–ой и i–ой ошибками* (i=[1,30], также смотри примечание в п.3), в соответствии с:
   1. равномерным законом распределения в интервале [0,20]; при этом cредний интервал между ошибками будет mравн = 10, СКО sравн = 20/(2\*sqrt(3)) = 5.8 .
   2. экспоненциальным законом распределения: W(y) = b\*exp(-b\*y), y>=0, c параметром b=0.1 и соответственно mэксп=sэксп= 1/b=10. Значения случайной величины Y с экспоненциальным законом распределения с параметром «b» можно получить по значениям случайной величины t, равномерно распределенной в интервале [0,1], по формуле [1]: Y = -ln(t) / b
   3. релеевским законом распределения: W(y) = (y/c^2)\*exp(-y^2/(2\*c^2)), y>=0, c параметром c=8.0 и соответственно mрел = c\*sqrt(π/2), sрел= c\*sqrt(2-π/2). Значения случайной величины Y с релеевским законом распределения с параметром «с» можно получить по значениям случайной величины t, равномерно распределенной в интервале [0,1], по формуле [1]: Y = с \* sqrt(-2\*ln(t)).
2. Каждый из 3-х массивов {Xi} интервалов времени между соседними ошибками упорядочить по возрастанию.
3. Для каждого из 3-х массивов {Xi}оценить значение первоначального числа ошибок в программе B. При этом для каждого закона использовать 100%, 80% и 60% входных данных (то есть в массивах {Хi} использовать n = 30, 24 и 18 элементов).

*Примечание*: для каждого значения n следует генерировать и сортировать новые массивы.

1. Если B>n, оценить значения средних времен Xj , j=n+1,n+2…, n+k до обнаружения k<= 5 следующих ошибок и общее время на выполнение тестирования.
2. Результаты вычислений представить в виде двух таблиц, одна из которых содержит оценки первоначального числа ошибок, а другая – оценки полных времен проведения тестирования - для разных законов распределения времен между отказами и разного числа используемых данных.
3. Сравнить и объяснить результаты, полученные для различных законов распределения времени между соседними отказами и различного числа используемых для анализа данных.

**Ход работы.**

1. Равномерный закон распределения

import numpy

numbers = numpy.random.uniform(0,20,30)

numbers.sort()

for i in range(30):

print(numbers[i])

|  |  |  |  |
| --- | --- | --- | --- |
|  | Равномерное 100 | Равномерное 80 | Равномерное 60 |
| 1 | 0.017 | 0.288 | 0.334 |
| 2 | 0.285 | 0.682 | 0.782 |
| 3 | 1.485 | 1.35 | 2.89 |
| 4 | 1.669 | 2.953 | 5.957 |
| 5 | 1.89 | 3.703 | 6.015 |
| 6 | 1.99 | 4.88 | 6.698 |
| 7 | 2.145 | 7.091 | 7.695 |
| 8 | 2.368 | 8.091 | 10.224 |
| 9 | 2.469 | 8.401 | 11.483 |
| 10 | 3.357 | 9.656 | 12.172 |
| 11 | 3.357 | 11.405 | 13.448 |
| 12 | 7.093 | 11.53 | 14.991 |
| 13 | 7.316 | 11.848 | 15.378 |
| 14 | 7.393 | 12.187 | 17.375 |
| 15 | 7.593 | 12.227 | 18.059 |
| 16 | 11.033 | 13.584 | 18.863 |
| 17 | 11.496 | 13.701 | 19.395 |
| 18 | 11.505 | 14.456 | 19.654 |
| 19 | 11.543 | 16.387 |  |
| 20 | 11.69 | 18.195 |  |
| 21 | 12.684 | 18.378 |  |
| 22 | 13.164 | 18.497 |  |
| 23 | 13.601 | 18.964 |  |
| 24 | 13.784 | 19.474 |  |
| 25 | 16.93 |  |  |
| 26 | 17.457 |  |  |
| 27 | 18.847 |  |  |
| 28 | 19.361 |  |  |
| 29 | 19.452 |  |  |
| 30 | 19.573 |  |  |

**Для n = 30**

Проверка существования максимума: 21,532 > 15.5

Поиск m

|  |  |  |  |
| --- | --- | --- | --- |
| m | f | g | |f - g| |
| 31 | 3.99499 | 3.168733957 | 0.8262531744 |
| 32 | 3.02725 | 2.866012879 | 0.1612323166 |
| **33** | **2.5585** | **2.616088136** | **0.0575929404** |
| 34 | 2.25546 | 2.40625558 | 0.1507906876 |

Минимум при m = 33, B = m - 1=32

Среднее время

|  |  |
| --- | --- |
| i | X |
| 31 | 26.04501 |
| 32 | 34.72668006 |

Время до полного завершения тестирования: 60.7717

Полное время тестирования: 333.316

**Для n = 24**

Проверка существования максимума: 16.3 > 12.5

Поиск m

|  |  |  |  |
| --- | --- | --- | --- |
| m | f | g | |f - g| |
| 25 | 3.73429 | 2.760614506 | 0.9736770051 |
| 26 | 2.77596 | 2.475830595 | 0.3001275824 |
| 27 | 2.31596 | 2.2443086 | 0.0716495781 |
| **28** | **2.02109** | **2.052384279** | **0.0312978964** |
| 29 | 1.80812 | 1.890699223 | 0.0825758027 |

Минимум при m = 28, B = m - 1 = 27

Среднее время

|  |  |
| --- | --- |
| i | X |
| 25 | 41.89092111 |
| 26 | 62.83638167 |
| 27 | 125.6727633 |

Время до полного завершения тестирования: 230.4000661284

Полное время тестирования: 488.329

**Для n = 18**

Проверка существования максимума: 12.375 > 10.5

Поиск m

|  |  |  |  |
| --- | --- | --- | --- |
| m | f | g | |f - g| |
| 19 | 3.43955 | 2.717118669 | 0.7224338532 |
| 20 | 2.49511 | 2.360759565 | 0.1343485136 |
| **21** | **2.04774** | **2.087037668** | **0.039298011** |
| 22 | 1.76441 | 1.8702 | 0.10579 |

Минимум при m = 21, B = m - 1 = 20

Среднее время

|  |  |
| --- | --- |
| i | X |
| 19 | 48.2532 |
| 20 | 96.5056 |

Время до полного завершения тестирования: 144.76

Полное время тестирования: 346.172

1. Экспоненциальный закон распределения

import numpy

numbers = numpy.random.exponential(10, 30)

numbers.sort()

for i in range(30):

print(numbers[i])

|  |  |  |  |
| --- | --- | --- | --- |
| i | Экспонента 100 | Экспонента 80 | Экспонента 60 |
| 1 | 0.564 | 0.306 | 0.767 |
| 2 | 0.804 | 0.940 | 2.113 |
| 3 | 1.255 | 1.076 | 2.424 |
| 4 | 1.339 | 1.627 | 3.246 |
| 5 | 1.397 | 3.826 | 3.977 |
| 6 | 3.209 | 3.888 | 4.197 |
| 7 | 3.212 | 4.002 | 4.442 |
| 8 | 3.496 | 4.726 | 6.346 |
| 9 | 3.833 | 5.508 | 7.145 |
| 10 | 3.881 | 6.274 | 7.451 |
| 11 | 4.141 | 6.673 | 11.199 |
| 12 | 4.706 | 6.702 | 11.434 |
| 13 | 4.749 | 7.714 | 11.618 |
| 14 | 5.695 | 8.076 | 16.621 |
| 15 | 5.852 | 10.444 | 22.346 |
| 16 | 6.228 | 10.592 | 26.762 |
| 17 | 6.564 | 12.851 | 30.480 |
| 18 | 6.776 | 15.255 | 36.644 |
| 19 | 6.925 | 18.598 |  |
| 20 | 7.138 | 22.324 |  |
| 21 | 8.989 | 22.982 |  |
| 22 | 9.560 | 28.436 |  |
| 23 | 12.531 | 30.811 |  |
| 24 | 13.820 | 34.225 |  |
| 25 | 14.052 |  |  |
| 26 | 14.597 |  |  |
| 27 | 29.372 |  |  |
| 28 | 31.791 |  |  |
| 29 | 34.416 |  |  |
| 30 | 45.776 |  |  |

**Для n = 30**

Проверка существования максимума: 23.273 > 15.5

Поиск m

|  |  |  |  |
| --- | --- | --- | --- |
| m | f | g | |f - g| |
| **31** | **3.99499** | **3.8826017061** | **0.1123854249** |
| 32.000 | 3.02725 | 3.437695021 | 0.4104498255 |

Минимум при m = 31, B = m - 1 = 30

Среднее время

|  |  |
| --- | --- |
| i | X |

Время до полного завершения тестирования: 0

Полное время тестирования: 296.670

**Для n = 24**

Проверка существования максимума: 18.132 > 12.5

Поиск m

|  |  |  |  |
| --- | --- | --- | --- |
| m | f | g | |f - g| |
| 25 | 3.77596 | 3.4948708139 | 0.2810873639 |
| **26.000** | **2.81596** | **3.0506380663** | **0.2346798886** |
| 27 | 2.35442 | 2.7066020924 | 0.3521823762 |

Минимум при m = 26, B = m - 1 = 25

Среднее время

|  |  |
| --- | --- |
| i | X |
| 25 | 87.8039 |

Время до полного завершения тестирования: 87.8039

Полное время тестирования: 355.662

**Для n = 18**

Проверка существования максимума: 13.748 > 10.5

Поиск m

|  |  |  |  |
| --- | --- | --- | --- |
| m | f | g | |f - g| |
| **19** | **3.43955** | **3.4272861569** | **0.0122663658** |
| 20.000 | 2.49511 | 2.8790930579 | 0.3839849797 |

Минимум при m = 19, B = m - 1 = 18

Среднее время

|  |  |
| --- | --- |
| i | X |

Время до полного завершения тестирования: 0

Полное время тестирования: 209.209

1. Релеевский закон распределения

import numpy

numbers = numpy.random.rayleigh(8, 30)

numbers.sort()

for i in range(30):

print(numbers[i])

|  |  |  |  |
| --- | --- | --- | --- |
| i | Релеевское 100 | Релеевское 80 | Релеевское 60 |
| 1 | 1.1114442195 | 2.7581015398 | 2.0466951642 |
| 2 | 1.3331616235 | 3.2324675271 | 4.0864488144 |
| 3 | 1.6695807272 | 3.7004411384 | 4.8822563263 |
| 4 | 3.8951358412 | 3.8666982765 | 5.3934901757 |
| 5 | 4.072814395 | 4.1830771545 | 5.3947891549 |
| 6 | 4.6596394404 | 4.8889654686 | 6.4094400065 |
| 7 | 5.0204047531 | 5.5697691136 | 6.8898468829 |
| 8 | 6.2348406978 | 6.5859418369 | 7.6636048118 |
| 9 | 6.3062186191 | 6.7075193058 | 7.6909608313 |
| 10 | 6.3474623154 | 7.0622446794 | 7.9890559452 |
| 11 | 6.3482272322 | 7.229309367 | 8.3902596525 |
| 12 | 6.8159137698 | 7.9177001344 | 8.567920585 |
| 13 | 7.3675891476 | 7.9443981036 | 9.2877580628 |
| 14 | 7.4767418915 | 9.1741120831 | 10.7876499917 |
| 15 | 7.6065027639 | 11.4745722275 | 11.2125672953 |
| 16 | 7.6734119909 | 12.1317814404 | 16.263834967 |
| 17 | 8.1177194007 | 12.3772020372 | 19.6444503342 |
| 18 | 8.3919813076 | 12.4563032301 | 20.9818411435 |
| 19 | 9.4841863996 | 13.5120301669 |  |
| 20 | 9.6524168897 | 13.7603650448 |  |
| 21 | 9.9907813425 | 14.3380016651 |  |
| 22 | 10.0800836703 | 14.3920958088 |  |
| 23 | 10.8325967215 | 16.4051399882 |  |
| 24 | 11.4846704026 | 17.9566756362 |  |
| 25 | 12.2836063628 |  |  |
| 26 | 15.0062680601 |  |  |
| 27 | 16.218744595 |  |  |
| 28 | 18.0657282599 |  |  |
| 29 | 18.2701066679 |  |  |
| 30 | 20.7880431105 |  |  |

**Для n = 30**

Проверка существования максимума: 20.106 > 15.5

Поиск m

|  |  |  |  |
| --- | --- | --- | --- |
| m | f | g | |f - g| |
| 31 | 3.99499 | 2.7538671069 | 1.241120024 |
| 32.000 | 3.02725 | 2.522328522 | 0.5049166735 |
| 33 | 2.55850 | 2.3267047317 | 0.2317904637 |
| **34.000** | **2.03488** | **2.159240867** | **0.1243642099** |
| 35 | 1.86345 | 2.0142647731 | 0.1508166875 |

Минимум при m = 34, B = m - 1 = 33

Среднее время

|  |  |
| --- | --- |
| i | X |
| 31 | 40.5398685305 |
| 32 | 60.8098027957 |
| 33 | 121.6196055914 |

Время до полного завершения тестирования: 222.9692769176

Полное время тестирования: 485.575

**Для n = 24**

Проверка существования максимума: 15.816 > 12.5

Поиск m

|  |  |  |  |
| --- | --- | --- | --- |
| m | f | g | |f - g| |
| 25 | 3.77596 | 2.6134473749 | 1.1625108029 |
| 26.000 | 2.81596 | 2.3568061707 | 0.4591520071 |
| 27 | 2.35442 | 2.146062301 | 0.2083574152 |
| 28.000 | 2.05812 | 1.9699140402 | 0.0882093797 |
| 29 | 1.84384 | 1.8204887737 | 0.0233489319 |
| **30.000** | **1.67832** | **1.6921341402** | **0.0138136759** |
| 31 | 1.54499 | 1.5806868804 | 0.0356997494 |

Минимум при m = 30, B = m - 1 = 29

Среднее время

|  |  |
| --- | --- |
| i | X |
| 25 | 18.5416668944 |
| 26 | 21.6319447102 |
| 27 | 25.9583336522 |
| 28 | 32.4479170653 |
| 29 | 43.2638894203 |

Время до полного завершения тестирования: 141.844

Полное время тестирования: 361.469

**Для n = 18**

Проверка существования максимума: 12.088 > 10.5

Поиск m

|  |  |  |  |
| --- | --- | --- | --- |
| m | f | g | |f - g| |
| 19 | 3.43955 | 2.6042649484 | 0.8352875742 |
| 20.000 | 2.49511 | 2.2751002857 | 0.2200077925 |
| **21.000** | **2.04774** | **2.0198077724** | **0.0279318848** |
| 22.000 | 1.76441 | 1.8160284213 | 0.0516220975 |

Минимум при m = 21, B = m - 1 = 20

Среднее время

|  |  |
| --- | --- |
| i | X |
| 19 | 40.2532381064 |
| 20 | 80.5064762129 |

Время до полного завершения тестирования: 121.484

Полное время тестирования: 285.067

1. Результаты
   1. Оценка первоначального числа ошибок

|  |  |  |  |
| --- | --- | --- | --- |
|  | Равномерный | Экспоненциальный | Релеевский |
| n = 30 | 32 | **30** | 33 |
| n = 24 | 27 | **25** | 29 |
| n = 18 | 20 | **18** | 20 |

* 1. Оценка полного времени проведения тестирования

|  |  |  |  |
| --- | --- | --- | --- |
|  | Равномерный | Экспоненциальный | Релеевский |
| n = 30 | 333.316 | **296.670** | 485.575 |
| n = 24 | 488.329 | **355.662** | 361.469 |
| n = 18 | 346.172 | **209.209** | 285.067 |

Экспоненциальный закон распределения показывает наилучшие результаты по двум оценкам сразу при любых входных данных, так как по предположению модели Джелински-Моранды время до следующего отказа программы распределено экспоненциально.

Релеевское распределение демонстрирует наихудшие результаты полного времени проведения тестирования при 60% и 80% входных данных, однако в плане оценки первоначального числа ошибок сравнимо с равномерным. При 100% входных данных наихудший результат показывает равномерное распределение.

**Вывод**

В ходе выполнения данной лабораторной работы было выполнено исследование показателей надежности программ, характеризуемых моделью обнаружения ошибок.