ОТЧЕТ

по лабораторной работе № 4

«Метод анализа иерархий»

по дисциплине

«ОСНОВЫ СИСТЕМНОГО АНАЛИЗА»

Выполнил студент группы ИС/б-22о

Горбенко К.Н.

Проверила:

Тлуховская Н.П.

* 1. ЦЕЛЬ РАБОТЫ

Углубление теоретических знаний в области системного анализа, приобретение навыков создания и описания иерархических структур, а также изучение понятий цель, критерий, альтернатива.

* 1. ЗАДАНИЕ НА РАБОТУ

Для **варианта № 22** задана следующая задача. Для организации наблюдения фирме нужно выбрать несколько телевизоров, выбор осуществляется в зависимости от технологий LED, LCD, ЖК.

Таблица 1 – Критерии альтернатив для выбора телевизора

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерии** | **LED** | **LCD** | **ЖК** |
| Яркость | 200 | 450 | 500 |
| Контрастность | 10000 | 40000 | 50000 |
| Время отклика матрицы | 4 | 8 | 4 |
| Угол обзора | 176 | 178 | 178 |
| Ресурс работы | 40000 | 60000 | 100000 |

* 1. ХОД РАБОТЫ

Составим пять матриц попарных сравнений для третьего уровня по отношению к критериям второго уровня.

Таблица 2 – Сравнение телевизоров с точки зрения яркости

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Яркость | LED | LCD | ЖК | Собственный вектор | x | Y |
| LED | 1 | 200/450 | 200/500 | 0.5625 | 0.174 | 0.521 |
| LCD | 450/200 | 1 | 450/500 | 1.2651 | 0.391 | 1.174 |
| ЖК | 500/200 | 500/450 | 1 | 1.406 | 0.435 | 1.3 |
| Сумма |  |  |  | 3.2336 | 0.99 | 2.9 |

*.*

*.*

*.*

Таблица 3 – Сравнение телевизоров с точки зрения контрастности

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Контрастность | LED | LCD | ЖК | Собственный вектор | x | Y |
| LED | 1 | 1/4 | 1/5 | 0.368 | 0.1 | 0.3 |
| LCD | 4/1 | 1 | 4/5 | 1.474 | 0.4 | 1.2 |
| ЖК | 5/1 | 5/4 | 1 | 1.842 | 0.5 | 1.5 |
| Сумма |  |  |  | 3.684 | 1 | 3 |

.

.

.

Таблица 4 – Сравнение телевизоров с точки зрения времени отклика матрицы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Время отклика матрицы | LED | LCD | ЖК | Собственный вектор | x | Y |
| LED | 1 | 4/8 | 4/4 | 0.794 | 0.250 | 0.75 |
| LCD | 8/4 | 1 | 8/4 | 1.587 | 0.5 | 1.5 |
| ЖК | 4/4 | 4/8 | 1 | 0.794 | 0.250 | 0.75 |
| Сумма |  |  |  | 3.175 | 1 | 3 |

0.75.

.

0.75.

Таблица 5 – Сравнение телевизоров с точки зрения угла обзора

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Угол обзора | LED | LCD | ЖК | Собственный вектор | x | Y |
| LED | 1 | 176/178 | 176/178 | 0.993 | 0.331 | 0.992 |
| LCD | 178/176 | 1 | 178/178 | 1.004 | 0.334 | 1.002 |
| ЖК | 178/176 | 178/178 | 1 | 1.004 | 0.334 | 1.002 |
| Сумма |  |  |  | 3.001 | 0.999 | 2.996 |

.

.

.

Таблица 6 – Сравнение телевизоров с точки зрения ресурса работы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ресурс работы | LED | LCD | ЖК | Собственный вектор | x | Y |
| LED | 1 | 4/6 | 4/10 | 0.644 | 0.2 | 0.6 |
| LCD | 6/4 | 1 | 6/10 | 0.965 | 0.3 | 0.917 |
| ЖК | 10/4 | 10/6 | 1 | 1.609 | 0.5 | 1.5 |
| Сумма |  |  |  | 3.218 | 1 | 3.017 |

.

.

.

Синтез приоритетов для уровня 3:

Таблица 7 – Вектора приоритетов для уровня 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вид матрицы | Яркость | Контрастность | Время отклика | Угол обзора | Ресурс работы |
| LED | 0.174 | 0.1 | 0.25 | 0.331 | 0.2 |
| LCD | 0.391 | 0.4 | 0.5 | 0.334 | 0.3 |
| ЖК | 0.435 | 0.5 | 0.25 | 0.334 | 0.5 |

Составим матрицу попарных сравнений на основе экспертных оценок:

Таблица 8 – Матрица попарных сравнений критериев

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Общее удовлетворение телевизором | 1 | 2 | 3 | 4 | 5 |
| Яркость (1) | 1 | 5 | 4 | 5 | 3 |
| Контрастность (2) | 1/5 | 1 | 1/2 | 2 | 1/2 |
| Время отклика матрицы (3) | 1/4 | 2 | 1 | 2 | 1/4 |
| Угол обзора (4) | 1/5 | 1/2 | 1/2 | 1 | 1/2 |
| Ресурс работы (5) | 1/3 | 2 | 4 | 2 | 1 |

Вычислим приоритеты критериев:

*.*

*.*

*.*

*.*

*.*

*.*

*.*

*.*

*.*

*.*

*.*

Вычислим глобальные приоритеты:

*.*

*.*

*.*

.

На основе проведенных вычислений вариант c телевизором ЖК получил наивысший вес и, следовательно, является наиболее оптимальным.

* 1. ТЕКСТ ПРОГРАММЫ

import numpy

def nth\_degree\_root(number, degree):

"""

Function to get the root of n-th degree from number

Args:

- number (float) : radical expression

- degree (float) : root degree

Return:

- N-th degree root from number

"""

return number \*\* (1 / degree)

def get\_criteria\_priorities(comparison\_matrix):

"""

Function to get the priorities on each criteria based on

comparison matrix (m x m) prepared by experts

Args:

- comparison\_matrix (numpy.matrix) : criterias comparison matrix

Return:

- list of m criteria priorities

"""

criteria\_priorities = []

matrixSum = comparison\_matrix.sum()

for row in comparison\_matrix.tolist():

criteria\_priorities.append(sum(row) / matrixSum)

return criteria\_priorities

def get\_alternatives\_criterias\_values(criteria\_characteristics):

"""

Function to get matrix of respective estimations

of all alternatives for one specific criteria represented by vector

Args:

- criteria\_characteristics (list) : list of characteristics for one criteria

Return:

- list of alternative estimation for one characteristic

"""

N = len(criteria\_characteristics)

comparison\_matrix = []

priorities = []

for i in range(0, N):

row = []

for j in range(0, N):

row.append(criteria\_characteristics[i] / criteria\_characteristics[j])

comparison\_matrix.append(row)

for i in range(0, N):

priorities.append(nth\_degree\_root(numpy.prod(comparison\_matrix[i]), N))

priorities\_sum = sum(priorities)

for i in range(0, N):

priorities[i] = priorities[i] / priorities\_sum

return priorities

def enter\_matrix\_from\_file(rows, columns, path):

"""

Function to input matrix from file. Elements of matrix

in file should be separated by one space. Use rows/columns

args to limit dimentions of resulting matrixes.

Args:

- rows (int) : number of rows in resulting matrix

- columns (int) : number of columns in resulting matrix

Return:

- rows by columns list of lists representation of matrix

"""

matrix = []

file = open(path)

for i in range(0, rows):

matrix.append(

list(map(float, file.readline().split()[0:columns]))

)

return matrix

def get\_alternatives\_total\_estimation(criterias, alternatives, criterias\_priorities\_matrix, characteristics\_matrix):

"""

Function to count total estimation value of each alternative

Args:

criterias (int) : criterias number

alternatives (int) : alternatives number

criterias\_priorities\_matrix (list) : matrix of criterias priorities

characteristics\_matrix (list) : matrix of characteristics

"""

characteristics = numpy.matrix(characteristics\_matrix)

criterias\_priorities = get\_criteria\_priorities(numpy.matrix(criterias\_priorities\_matrix))

alternatives\_respective\_values = []

alternatives\_total\_estimation = []

for row in characteristics.tolist():

alternatives\_respective\_values.append(get\_alternatives\_criterias\_values(row))

for i in range(alternatives):

total = 0

for j in range(criterias):

total += alternatives\_respective\_values[j][i] \* criterias\_priorities[j]

alternatives\_total\_estimation.append(total)

return alternatives\_total\_estimation

characteristics\_file\_path = "characteristics.txt"

criterias\_priorities\_file\_path = "criterias\_priorities.txt"

print("Enter number of criterias:")

criterias\_number = int(input())

print("Enter number of alternatives")

alternatives\_number = int(input())

characteristics\_matrix = enter\_matrix\_from\_file(criterias\_number, alternatives\_number, characteristics\_file\_path)

criterias\_comparison\_matrix = enter\_matrix\_from\_file(criterias\_number, criterias\_number, criterias\_priorities\_file\_path)

alternatives\_estimation = get\_alternatives\_total\_estimation(criterias\_number, alternatives\_number, criterias\_comparison\_matrix, characteristics\_matrix)

print (alternatives\_estimation)

* 1. РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

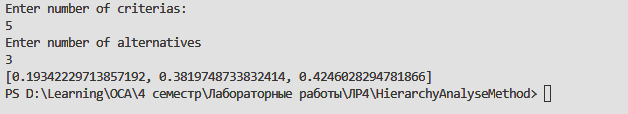


Рисунок 1 – Результат работы программы

Результат работы программы совпал с результатами практического расчета.

* 1. ВЫВОД

В ходе лабораторной работы была решена задача выбора одной из альтернатив по заданным приоритетам критериев методом анализа иерархий. Разработанная программная реализация метода работает в пределах множества задач, подходящих под описание доминантной иерархической структуры, то есть не подходит для иерархий, содержащих более трех уровней.

Для данной задачи наиболее подходящей альтернативой оказался ЖК-телевизор в основном из-за того, что он имеет наибольшие показатели критериев яркости и ресурса работы, которые эксперты оценили высокими приоритетами.