Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

По дисциплине: “Основы системного анализа”

Лабораторная работа №4

“Метод анализа иерархий”

Вариант 3

Выполнил:

ст.гр. ИС/б-22

Долженко И.А.

Проверил:

Гончаренко Д. Г.

Севастополь

2019

1 ЦЕЛЬ РАБОТЫ

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

2 ПОСТАНОВКА ЗАДАЧИ

1. Решить задачу методом МАИ вручную.
2. Написать программу на языке программирования python которая решает МАИ задачу любой размерности.

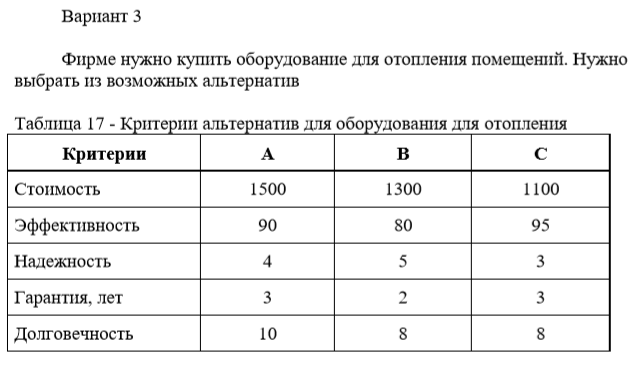


Таблица 2 – Экспертные оценки приоритетов характеристик

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| Стоимость | **1** | 5 | 4 | 5 | 3 |
| Эффективность | 1/5 | **1** | 1/2 | 2 | 1/2 |
| Надежность | 1/4 | 2 | **1** | 2 | 1/4 |
| Гарантия, лет | 1/5 | 1/2 | 1/2 | **1** | 1/2 |
| Долговечность | 1/3 | 2 | 4 | 2 | **1** |

ХОД РАБОТЫ

1. Код программы:

import numpy

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

return number \*\* (1 / degree)

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

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):

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):

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):

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)

Результат:

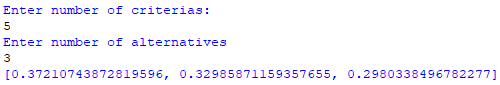


Рисунок 1 – Результат работы программы

Промежуточные результаты:

Таблица 3 – Сравнение вариантов по стоимости

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| A | 1 | 1.154 | 1.364 |
| B | 0.867 | 1 | 1.182 |
| C | 0.734 | 0.846 | 1 |

Таблица 4 – Сравнение вариантов по эффективности

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| A | 1 | 1.125 | 0.947 |
| B | 0.889 | 1 | 0.842 |
| C | 1.056 | 1.188 | 1 |

Таблица 5 – Сравнение вариантов по надежности

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| A | 1 | 0.8 | 1.333 |
| B | 1.25 | 1 | 1.667 |
| C | 0.75 | 0.6 | 1 |

Таблица 6 – Сравнение вариантов по гарантии

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| A | 1 | 1.5 | 1 |
| B | 0.667 | 1 | 0.667 |
| C | 1 | 1.5 | 1 |

Таблица 7 – Сравнение вариантов по долговечности

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| A | 1 | 1.25 | 1.25 |
| B | 0.8 | 1 | 1 |
| C | 0.8 | 1 | 1 |

Таблица 8 – Вектор приоритетов для уровня 2

|  |  |
| --- | --- |
| Характеристика | Вектор приоритетов |
| Стоимость | 0.489 |
| Эффективность | 0.099 |
| Гарантия | 0.118 |
| Надежность | 0.075 |
| Долговечность | 0.219 |

Таблица 9 – Вектора приоритетов для уровня 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вариант | Стоимость | Эффективность | Гарантия | Надежность | Долговечность |
| A | 0.385 | 0.340 | 0.333 | 0.375 | 0.385 |
| B | 0.334 | 0.302 | 0.417 | 0.25 | 0.308 |
| C | 0.282 | 0.358 | 0.250 | 0.375 | 0.308 |

ВЫВОДЫ

В ходе лабораторной работы изучен метод анализа иерархий (МАИ). МАИ позволяет понятным и рациональным образом структурировать сложную проблему принятия решений в виде иерархии, сравнить и выполнить количественную оценку альтернативных вариантов решения. На основе изученного метода разработана программа на языке Python. С помощью разработанной программы определена лучшая из альтернатив и проверены аналитические вычисления.