# Лабораторная работа №3

Метод решающих матриц

## Цель работы

Углубление теоретических знаний в области системного анализа, исследование способов оценки сложных систем.

## Постановка задачи

Приобрести навыки оценки сложных систем с помощью Метода решающих матриц.

## Вариант задания

(Вариант 1) Оцените влияние факторов нижнего уровня на проектирование всей системы в целом. Связи между уровнями указаны для каждого варианта отдельно. Веса первого уровня для всех вариантов едины:

a=[0.4; 0.2; 0.3; 0.1]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  |  |  |  | |  | 0.6 |  | 0.4 |  | |  |  |  |  | 1.0 | |  | 0.2 | 0.7 | 0.1 |  | |  | 0.9 |  |  | 0.1 | |  | 0.2 | 0.4 | 0.2 | 0.2 | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  | |  | 0.5 |  | 0.1 |  | 0.4 | |  |  | 0.5 | 0.3 | 0.2 |  | |  | 0.6 |  |  | 0.4 |  | |  |  | 0.4 | 0.3 |  | 0.3 | | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  |  |  |  | |  | 0.6 | 0.2 |  | 0.2 | |  | 0.3 |  | 0.2 | 0.5 | |  |  | 0.5 | 0.3 | 0.2 | |

## Ход выполнения работы

### Проведение вычислений

В соответствие с матрицами была построена иерархия декомпозиции проблемы (Рисунок 3.1):

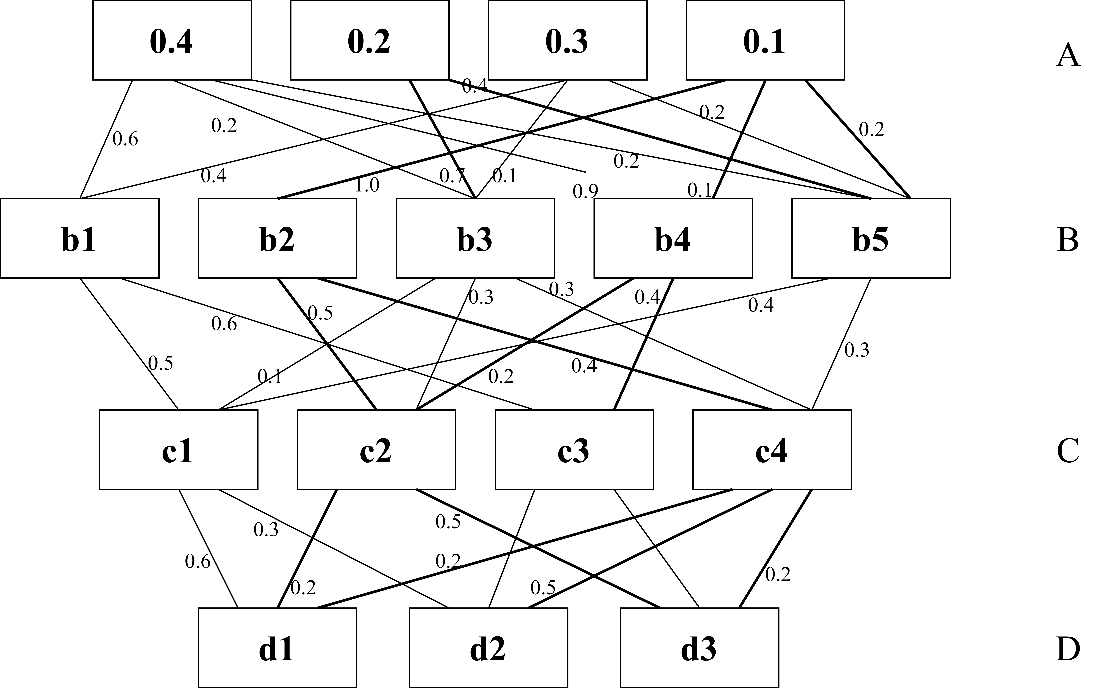


Рисунок 3.1 – Иерархия декомпозиции проблемы с весами

Определение относительного веса второго уровня:

Нормирование элементов :

Получены веса второго уровня:

Определение относительного веса третьего уровня:

Нормирование элементов :

Получены веса третьего уровня:

Определение относительного веса четвёртого уровня:

Нормирование элементов :

Получены веса четвёртого уровня:

Откуда следует, что вариант оказывает большее влияние на проектирование всей системы.

### Текст программы

Ниже представлена программа, написанная для интерпретатора Python 3.10.

#!/bin/python3

from os import system

from math import fsum

def clear\_screen (): # Очистить экран консоли

system("cls")

def print\_array (arr: list, arr\_name: str): # Печать массива

print ("%s = [" % arr\_name, end="")

for i in range (len(arr)):

print (("%5.2f, " if i + 1 != len (arr) else "%5.2f") % arr[i], end="")

print ("]")

def print\_matrix (m: list, highest\_level: str, lowest\_level: str): # Вывод матрицы на экран

lines, cols = len(m), len(m[0])

print(" ", end="")

for i in range(cols):

print("%.1s%-2d" % (highest\_level, i + 1), end=" ")

print()

for i in range(lines):

print("%.1s%-2d" % (lowest\_level, i + 1), end=" ")

for j in range(cols):

print("%5.2f" % m[i][j], end=" ")

print()

count = {

"a": 0,

"b": 0,

"c": 0

}

matrix = {

"ab": [],

"bc": [],

"cd": []

}

level = {

"a": [],

"b": [],

"c": [],

"d": []

}

tmp = 0

def main ():

clear\_screen()

# Ввод количества элементов на уровнях a, b, c, d

for level\_char in ("a", "b", "c", "d"):

count[level\_char] = int( input(f"Введите количесво элементов на уровне \"{level\_char}\": ") )

# Создание шаблона значений

for level\_char in ("a", "b", "c", "d"):

level[level\_char] = [0 for i in range(count[level\_char])]

# Ввод значений элементов на уровне a

for i in range(count["a"]):

clear\_screen ()

print\_array(level["a"], "a")

level["a"][i] = float( input("Введите значение веса a[%02d]: " % (i + 1)) )

# Создание шаблонов матриц

for level\_1, level\_2 in (("a", "b"), ("b", "c"), ("c", "d")):

matrix[level\_1 + level\_2] = [[0 for i in range(count[level\_1])] for j in range(count[level\_2])]

# Заполнение матриц весов

for level\_1, level\_2 in (("a", "b"), ("b", "c"), ("c", "d")):

for col in range(count[level\_1]):

for line in range(count[level\_2]):

clear\_screen()

print\_matrix(matrix[level\_1 + level\_2], level\_1, level\_2)

matrix[level\_1 + level\_2][line][col] = float(input(f"Введите {level\_1}{level\_2}[{line}][{col}]: "))

# Подсчёт значений

for level\_1, level\_2 in (("a", "b"), ("b", "c"), ("c", "d")):

for i in range(count[level\_2]):

level[level\_2][i] = fsum([(level[level\_1][j] \* matrix[level\_1 + level\_2][i][j]) for j in range(count[level\_1])])

sum = fsum(level[level\_2])

for i in range(count[level\_2]):

level[level\_2][i] = level[level\_2][i] / sum

clear\_screen ()

# Выводы

print ("Заданы матрицы весов:")

for level\_1, level\_2 in (("a", "b"), ("b", "c"), ("c", "d")):

print\_matrix(matrix[level\_1 + level\_2], level\_1, level\_2)

print()

print("Получены значения:")

for level\_char in ("a", "b", "c", "d"):

print\_array(level[level\_char], level\_char)

print( "Таким образом, вариант d%d оказывает наибольшее влияние на проектирование всей системы." % (level["d"].index(max(level["d"])) + 1) )

# Запуск основной программы

main ()

### Тестовые примеры

Рисунок 3.2 демонстрирует выполнение программы, если были введены исходные данные, соответствующие выбранному варианту.

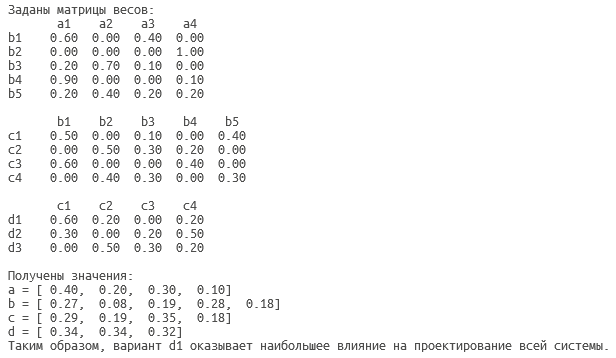


Рисунок 3.2 – Результат выполнения программы

## Вывод

При выполнении данной лабораторной работы были получены навыки оценки сложных систем с помощью Метода решающих матриц. Полученные во время выполнения лабораторной работы навыки помогут в дальнейшей жизни при необходимости провести системный анализ в различных областях.