Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий

Кафедра ИС

**ОТЧЁТ**

по лабораторной работе №3

МЕТОД РЕШАЮЩИХ МАТРИЦ

Выполнил:

ст. гр. ИС/б-21-2-о

Мовенко К. М.

Проверил:

Хохлов В. В.

Севастополь

2023

Цель работы

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

Задачи

1. На основе заданного варианта задания построить взаимосвязь всех уровней системы;
2. Вычислить влияние вариантов самого нижнего уровня на проектирование всей системы;
3. Написать программу на языке программирования Python (или др.), которая решает задачу методом решающих матриц любой размерности;
4. Дополнительное задание: написать программу на языке программирования Python (или др.), которая рисует в консоли граф на основе введённых матриц;

Вариант задания

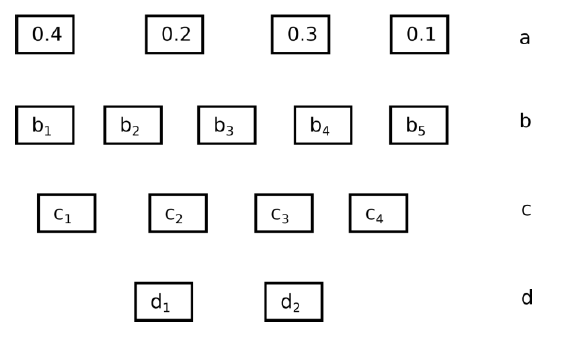


Рисунок 1 – Иерархия декомпозиции проблемы

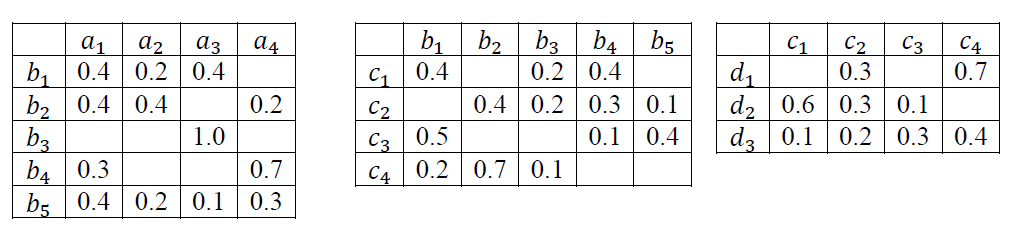


Рисунок 2 – Связи между уровнями иерархии (Вариант − 3)

Ход работы

В соответствии с матрицами связей, обозначенными в варианте задания (Рисунок 2), была построена полная иерархия декомпозиции проблемы со всеми связями между уровнями (Рисунок 3).

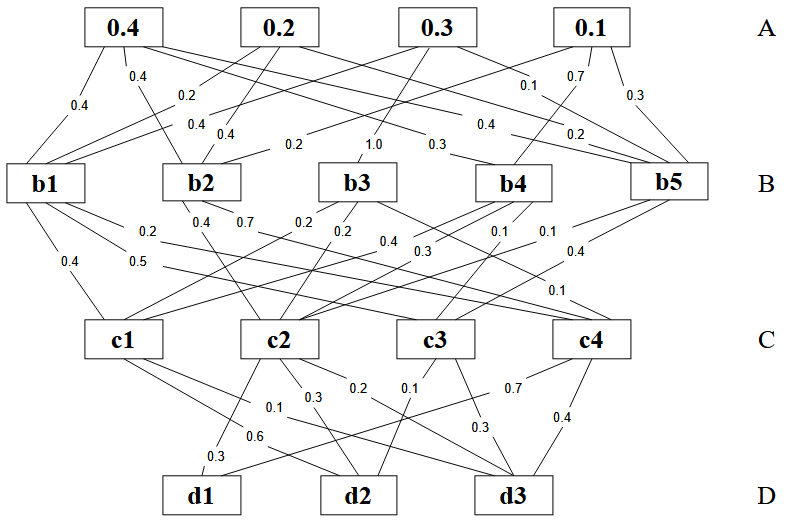


Рисунок 3 – Иерархия декомпозиции проблемы со связями

Далее были начаты расчёты.

Определение весов уровня :

Нормирование элементов :

Определение весов уровня :

Нормирование элементов :

Определение весов уровня:

Нормирование элементов :

Таким образом вариант оказывает более существенное влияние на проектирование всей системы.

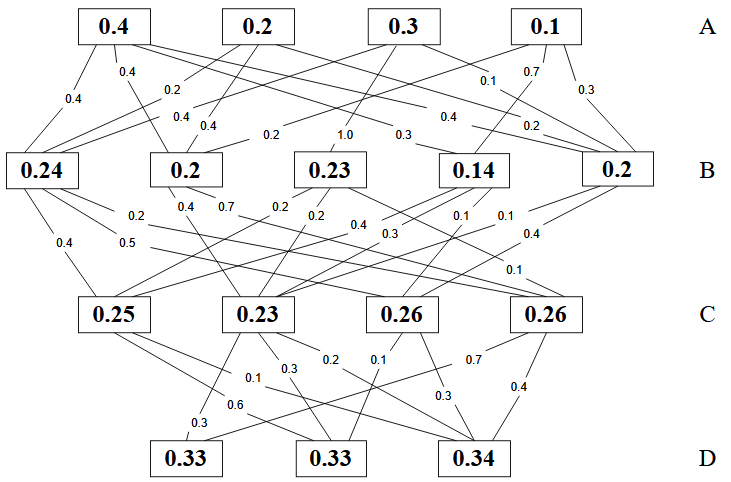


Рисунок 4 – Заполнение иерархии декомпозиции проблемы

Также была запущена программа, проводящая те же самые вычисления автоматически. Результаты её выполнения совпали с результатами ручных вычислений (Рисунок 5).

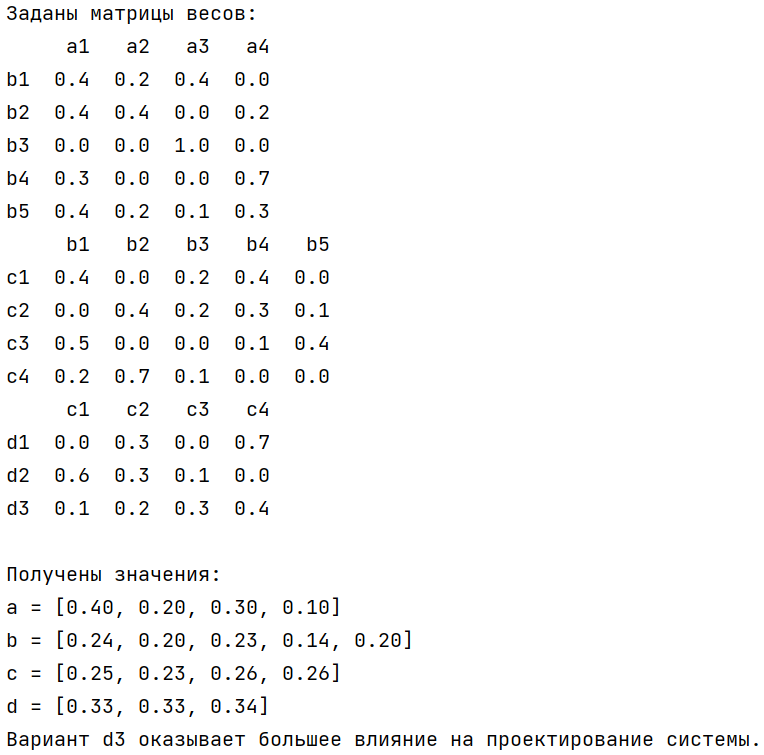


Рисунок 5 – Результат выполнения программы

Текст программы

Листинг 1 – Программа решения задачи методом решающих матриц

from os import system  
  
*# Печать массива*def print\_array(arr, arr\_name):  
 print("%s = [" % arr\_name, end="")  
  
 for i in range(len(arr)):  
 print(("%4.2f, " if i + 1 != len(arr) else "%4.2f") % arr[i], end="")  
  
 print("]")  
  
*# Печать матрицы*def print\_matrix(m, lvl\_1, lvl\_2):  
 lines, cols = len(m), len(m[0])  
  
 print(' ' \* 5, end='')  
 for i in range(cols):  
 print(lvl\_1, i + 1, sep='', end=" ")  
 print()  
  
 for i in range(lines):  
 print(lvl\_2, i + 1, sep='', end=" ")  
 for j in range(cols):  
 print("%3.1f" % m[i][j], end=" ")  
 print()  
  
  
*# Значения весов на уровнях*weights = {  
 'a': [],  
 'b': [],  
 'c': [],  
 'd': []  
}  
  
*# Матрицы связей между уровнями*ties = {  
 ('a', 'b'): [],  
 ('b', 'c'): [],  
 ('c', 'd'): []  
}  
  
*# ВЫПОЛНЕНИЕ ПРОГРАММЫ  
  
# Ввод количества элементов уровней*for lvl in weights.keys():  
 weights\_n = int(input(f"Введите количесво весов уровня {lvl}: "))  
 weights[lvl] = [0 for i in range(weights\_n)]  
  
*# Ввод значений весов на уровне a*for i in range(len(weights['a'])):  
 system("cls")  
 print\_array(weights['a'], 'a')  
 weights['a'][i] = float(input(f"Введите значение веса a[{i + 1}]: "))  
  
*# Заполнение матриц связей между уровнями*for lvl\_1, lvl\_2 in ties.keys():  
 width = len(weights[lvl\_1])  
 height = len(weights[lvl\_2])  
 ties[(lvl\_1, lvl\_2)] = [[0 for i in range(width)] for j in range(height)]  
  
 for line in range(height):  
 for col in range(width):  
 system("cls")  
 print\_matrix(ties[(lvl\_1, lvl\_2)], lvl\_1, lvl\_2)  
 print(f"Введите связь {lvl\_1}{col + 1}-{lvl\_2}{line + 1}: ", end='')  
 ties[(lvl\_1, lvl\_2)][line][col] = float(input())  
  
*# Подсчёт весовых коэффициентов*for lvl\_1, lvl\_2 in ties.keys():  
 ties\_n = len(weights[lvl\_1])  
 weights\_n = len(weights[lvl\_2])  
  
 *# Подсчёт весов нижнего уровня* for i in range(weights\_n):  
 b = [(weights[lvl\_1][j] \* ties[(lvl\_1, lvl\_2)][i][j]) for j in range(ties\_n)]  
 weights[lvl\_2][i] = sum(b)  
  
 *# Нормирование элементов* weight\_sum = sum(weights[lvl\_2])  
 for i in range(weights\_n):  
 weights[lvl\_2][i] = weights[lvl\_2][i] / weight\_sum  
  
system("cls")  
  
*# Выводы*print("Заданы матрицы весов:")  
for lvl\_1, lvl\_2 in ties.keys():  
 print\_matrix(ties[(lvl\_1, lvl\_2)], lvl\_1, lvl\_2)  
 print()  
  
print("Получены значения:")  
for lvl in weights.keys():  
 print\_array(weights[lvl], lvl)  
  
max\_d = weights['d'].index(max(weights['d'])) + 1  
print(f"Вариант d{max\_d} оказывает большее влияние на проектирование системы.")

Контрольные вопросы

1. В чем заключается суть метода решающих матриц?

Суть метода решающих матриц заключается в создании иерархической структуры, в которой проблема разбивается на более мелкие подзадачи и критерии, которые затем сравниваются попарно в отношении их значимости и вклада в решение общей проблемы. Затем используются числовые методы для определения весов критериев и принятия решений на основе этих весов.

1. В каких случаях необходимо производить разбиение проблемы на подзадачи?

* Когда проблема слишком сложна или объёмна для решения в единой форме;
* Когда проблема требует экспертизы в разных областях знаний;
* Когда проблема имеет иерархическую структуру или включает в себя несколько взаимосвязанных аспектов;
* Когда проблема может быть решена последовательно, при этом решение одной подзадачи может служить основой для решения следующей;

Вывод

В ходе работы были получены навыки оценки сложных систем с помощью метода решающих матриц. Изученная методика помогает проводить системный анализ в различных областях, связанных с решением объёмных задач.