МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федерально автономное бюджетное образовательное учреждение высшего образования

«Севастопольский государственный университет»

кафедра Информационных систем

Куркчи Ариф Эрнестович

Институт информационных технологий и управления в технических системах

курс 3 группа ИС/б-31-о

09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Теория принятия решений»

на тему «Исследование применения метода анализа иерархий для решения задачи выбора альтернатив»

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

ст. преподаватель   А. Ю. Дрозин

(должность) (подпись) (инициалы, фамилия)

Севастополь 2016

1. Цель работы

Исследовать применение аппарата метода анализа иерархий при принятии решений по выбору альтернатив.

2. Постановка задачи

Вариант 3

В процессе дипломного проектирования возникает необходимость выбора темы дипломного проекта. (дипломный руководитель предлагает несколько тем на выбор). Цель принятия решений состоит в выборе темы для дипломного проектирования из предлагаемого перечня. Характеристиками (критериями), соответствующими свойствам решений, являются: сложность материала, положенного в основу темы дипломного проект; наличие знаний по материалу, на основе которого реализуется дипломный проект; возможность использования знаний, полученных при дипломном проектировании по выбранной теме, в дальнейшей деятельности; наличие свободного времени для реализации выбранной темы дипломного проекта. Для реализации выбора необходимо сформировать требуемые матрицы парных сравнений и реализовать процедуру принятия решений. При этом для определения значений элементов собственных векторов матриц парных сравнений использовать третий из предложенных в Приложении А методов.

3. Текст программы

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1 | #include <iostream> | | 2 | #include <fstream> | | 3 | #include <cmath> | | 4 | #include <iomanip> | | 5 |  | | 6 | using namespace std; | | 7 |  | | 8 | *double*\* mult(*double* \*\**matrixA*, *double* \**matrixB*, *int* *n*) { | | 9 | *double* \*result = new *double*[n]; | | 10 | for (*int* i = 0; i < n; i++) { | | 11 | for (*int* j = 0; j < n; j++) { | | 12 | result[i] += matrixA[i][j] \* matrixB[j]; | | 13 | } | | 14 | } | | 15 | return result; | | 16 | } | | 17 |  | | 18 | *double*\* vectorW(*int* *n*, *double* \*\**m*, *double* &*l*, *double* &*IS*) { | | 19 | *double* \*w = new *double*[n]; | | 20 | *double* sum = 0; | | 21 | for (*int* i = 0; i < n; i++) { | | 22 | for (*int* j = 0; j < n; j++) { | | 23 | w[i] += m[j][i]; | | 24 | } | | 25 | w[i] = 1 / w[i]; | | 26 | sum += w[i]; | | 27 | } | | 28 | for (*int* i = 0; i < n; i++) { | | 29 | w[i] = w[i] / sum; | | 30 | } | | 31 | *double* \*w1 = mult(m, w, n); | | 32 | *double* \*w11 = new *double*[n]; | | 33 | l = 0; | | 34 | for (*int* i = 0; i < n; i++) { | | 35 | w11[i] = w1[i] / w[i]; | | 36 | l += w11[i] / n; | | 37 | } | | 38 | IS = (l - n) / (n - 1); | | 39 | return w; | | 40 | } | | 41 |  | | 42 | *double*\*\* read\_matrix(ifstream &*is*, *int* *n*) { | | 43 | *double* \*\*r = new *double*\*[n]; | | 44 | for(*int* i=0;i<n;i++) { | | 45 | r[i] = new *double*[n]; | | 46 | for(*int* j=0;j<n;j++) { | | 47 | is >> r[i][j]; | | 48 | } | | 49 | } | | 50 | return r; | | 51 | } | | 52 |  | | 53 | *void* print\_matrix(*int* *n*, *double* \*\**m*) { | | 54 | for(*int* i=0;i<n;i++) { | | 55 | for(*int* j=0;j<n;j++) { | | 56 | cout << setw(3) << m[i][j]; | | 57 | if(j < n-1) { | | 58 | cout << " "; | | 59 | } | | 60 | } | | 61 | cout << endl; | | 62 | } | | 63 | } | | 64 |  | | 65 | *double*\* calc(*int* *n*, *double* \*\**m*) { | | 66 | print\_matrix(n, m); | | 67 | *double* l, IS; | | 68 | *double*\* w = vectorW(n, m, l, IS); | | 69 | cout << "\tВектор W:" << endl; | | 70 | for(*int* i=0;i<n;i++) { | | 71 | cout << setw(4) << setprecision(2) << w[i] << " "; | | 72 | } | | 73 | cout << endl; | | 74 | cout << "Lmax = " << setw(4) << setprecision(2) << l << endl; | | 75 | cout << "IS   = " << setw(4) << setprecision(2) << IS << endl; | | 76 | cout << endl; | | 77 | return w; | | 78 | } | | 79 |  | | 80 | *int* main() { | | 81 | *int* n,m; | | 82 | ifstream is("input.txt"); | | 83 | is >> n >> m; | | 84 | *double* \*\*A1 = read\_matrix(is, n); | | 85 | *double* \*\*A21 = read\_matrix(is, m); | | 86 | *double* \*\*A22 = read\_matrix(is, m); | | 87 | *double* \*\*A23 = read\_matrix(is, m); | | 88 | *double* \*\*A24 = read\_matrix(is, m); | | 89 |  | | 90 | cout << "\tМатрица парных сравнений A1" << endl; | | 91 | *double* \*w1 = calc(n, A1); | | 92 |  | | 93 | *double* \*\*w2 = new *double*\*[4]; | | 94 | cout << "\tМатрица парных сравнений A21" << endl; | | 95 | w2[0] = calc(m, A21); | | 96 | cout << "\tМатрица парных сравнений A22" << endl; | | 97 | w2[1] = calc(m, A22); | | 98 | cout << "\tМатрица парных сравнений A23" << endl; | | 99 | w2[2] = calc(m, A23); | | 100 | cout << "\tМатрица парных сравнений A24" << endl; | | 101 | w2[3] = calc(m, A24); | | 102 |  | | 103 | *double* D\_max = 0; | | 104 | *int* D\_index = 0; | | 105 | for(*int* i=0;i<n;i++) { | | 106 | *double* D = 0; | | 107 | for(*int* j=0;j<4;j++) { | | 108 | D += w2[j][i] \* w1[j]; | | 109 | } | | 110 | if(D > D\_max) { | | 111 | D\_max = D; | | 112 | D\_index = i + 1; | | 113 | } | | 114 | cout << "D" << (i + 1) << " = " << D << endl; | | 115 | } | | 116 | cout << "Оптимальное: " << D\_index << endl; | | 117 |  | | 118 | return 0; | | 119 | } | |  |

4. Результаты

На рисунке 4.1 представлен скриншот демонстрирующий работу написанной программы.

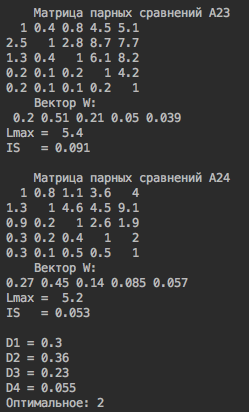
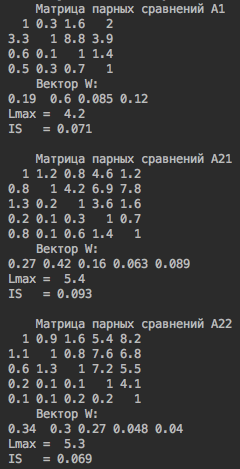


Рисунок 4.1 – Результат работы программы

Программа определила и указала на то, что оптимальным является решение 2.

Вывод

В ходе лабораторной работы было исследовано применение аппарата метода анализа иерархий при принятии решений по выбору альтернатив.