Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

По дисциплине: “Теория принятия решений”

Лабораторная работа №6

“Исследование применения метода анализа иерархий

для решения задачи выбора альтернатив”

Выполнил:

ст.гр. ИС/б-17-2-о

Черняев Н.Г.

Проверил:

Кротов К.В.

Севастополь

2020

1 ЦЕЛЬ РАБОТЫ

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

2 ПОСТАНОВКА ЗАДАЧИ

В процессе реализации дипломного или курсового проектов возникает необходимость в выборе языка программирования. (цель принятия решений– выбор языка программирования для реализации проекта). Характеристики, соответствующие свойствам альтернатив (решений), влияющие на цель принятия решений: наличие базовых знаний синтаксиса языка, соответствие языка современному уровню развития технологий программирования, сложность синтаксиса, имеющееся время на реализацию 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 метод 2 (Вариант 2):

1. Разделить элементы каждого столбца на сумму элементов этого столбца (нормализовать элементы каждого столбца)

2. Полученные элементы сложить для каждой строки (выполнить построчное суммирование полученных нормализованных элементов)

3. Разделить каждую из полученных сумм на число элементов в строке (процесс усреднения по нормализованным столбцам)

3 ХОД РАБОТЫ

1. Матрицы парных сравнений:

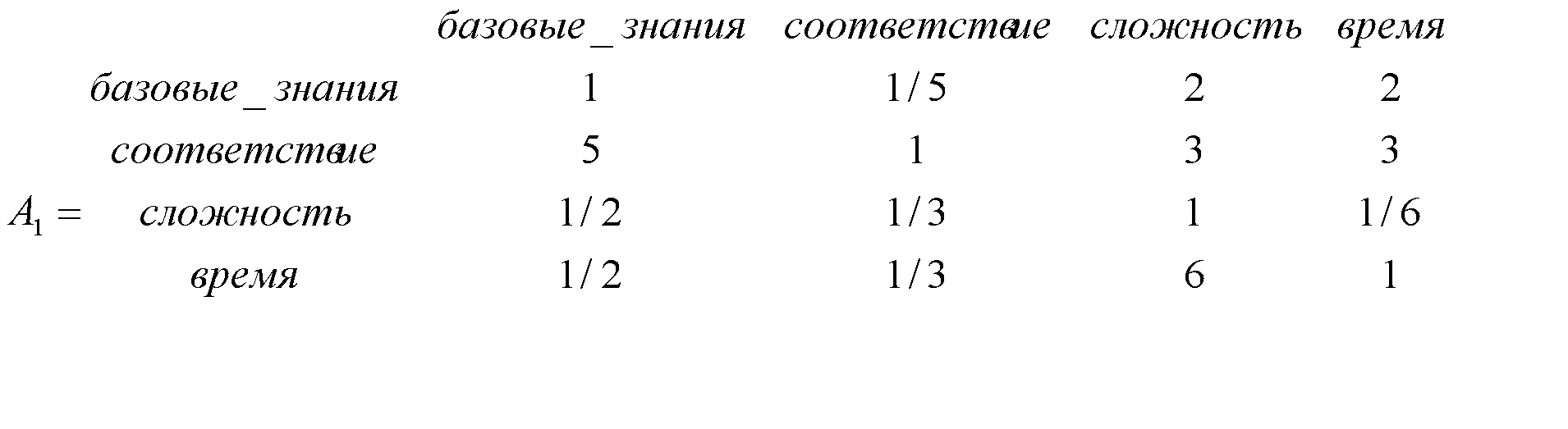


Рисунок 1 – Матрица А1 парных сравнений влияния характеристик альтернатив (решений) на общую цель принятия решений

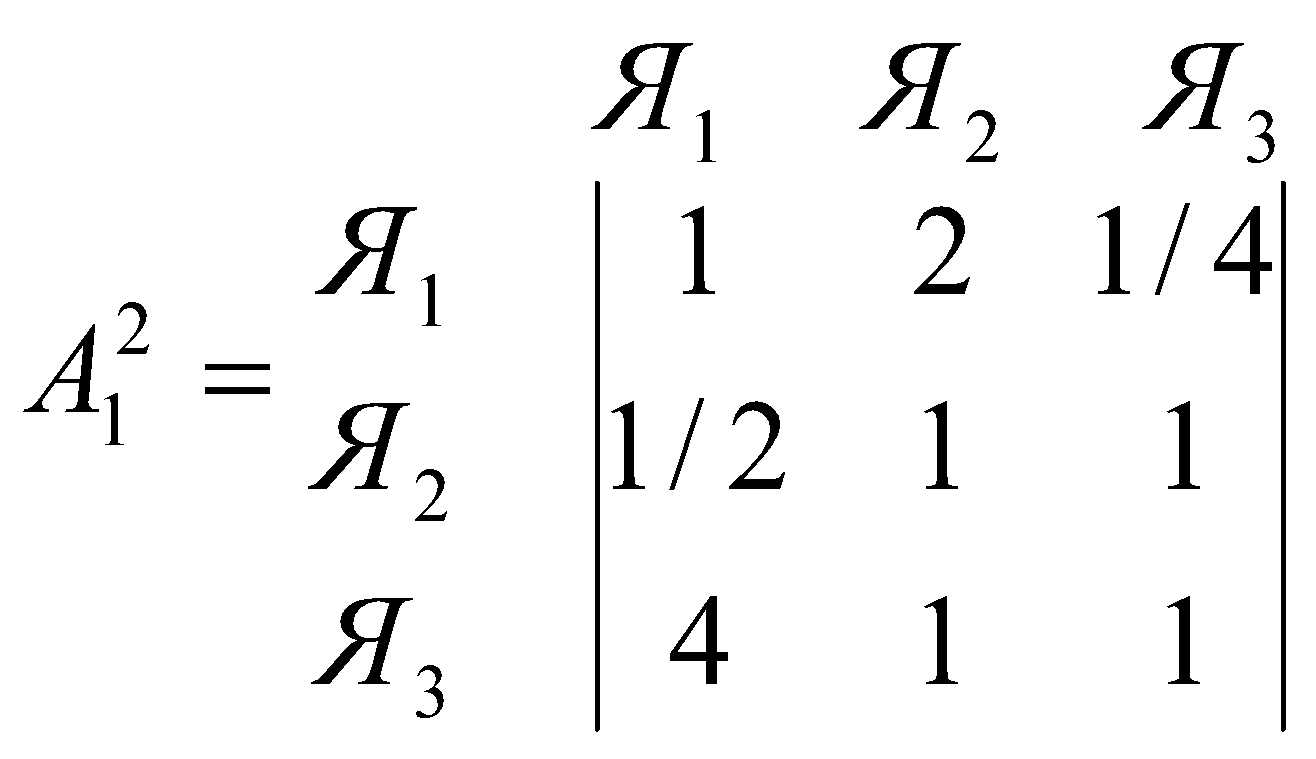


Рисунок 2 – Матрица А21 парных сравнений базовых знаний

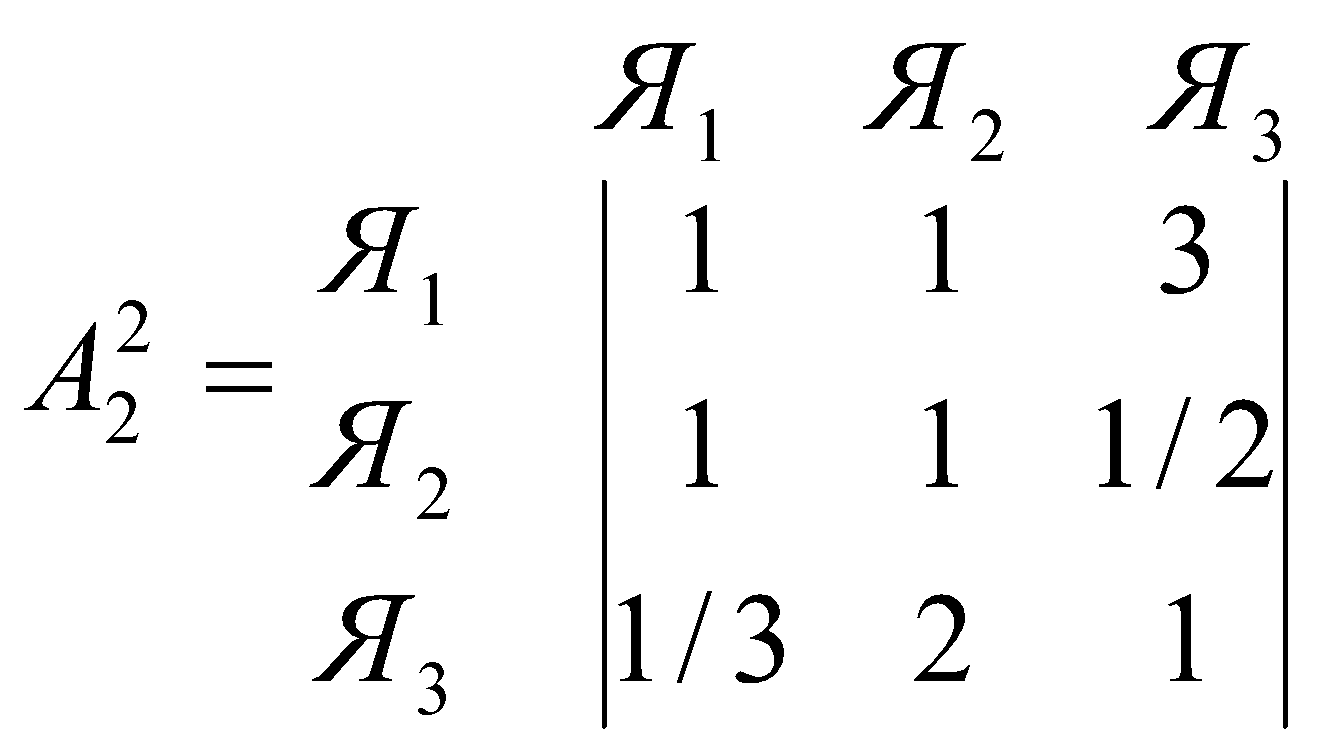


Рисунок 3 – Матрица А22 парных сравнений соответствий

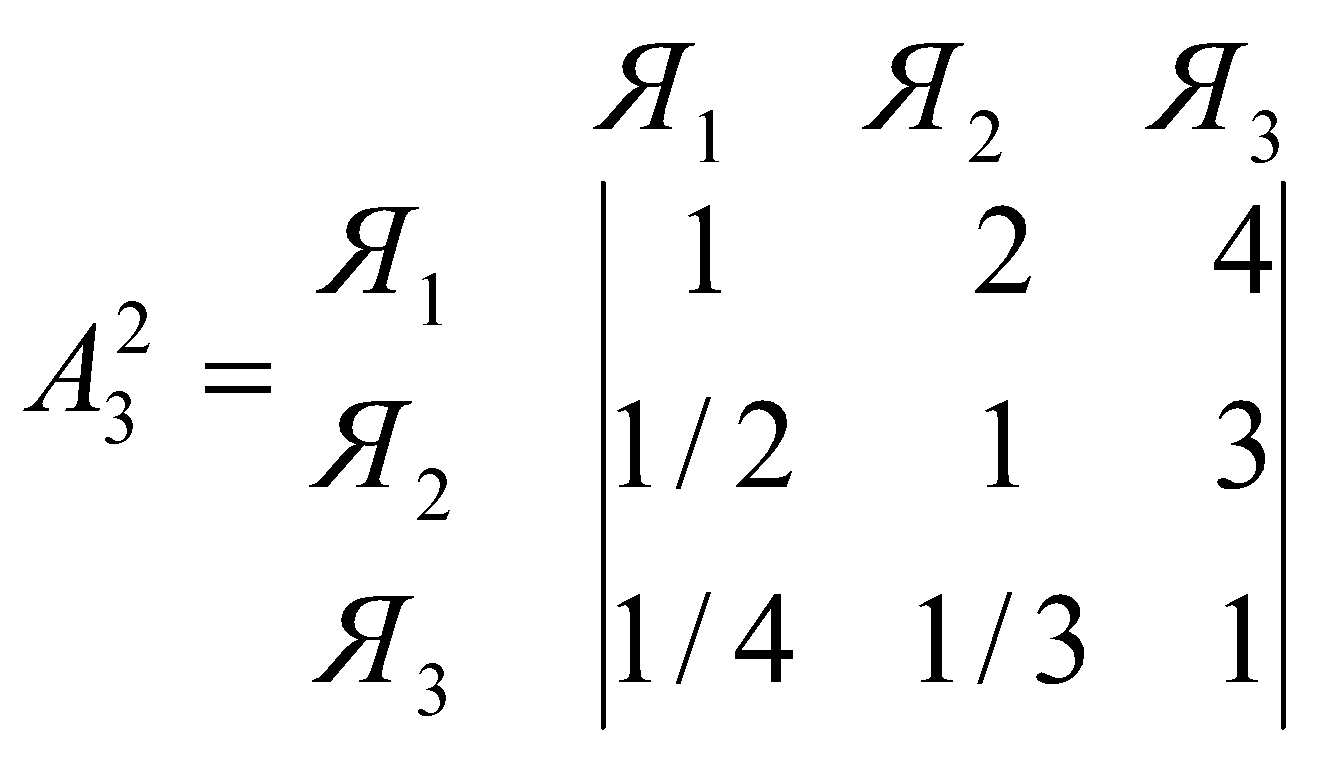


Рисунок 4 – Матрица А23 парных сравнений сложности

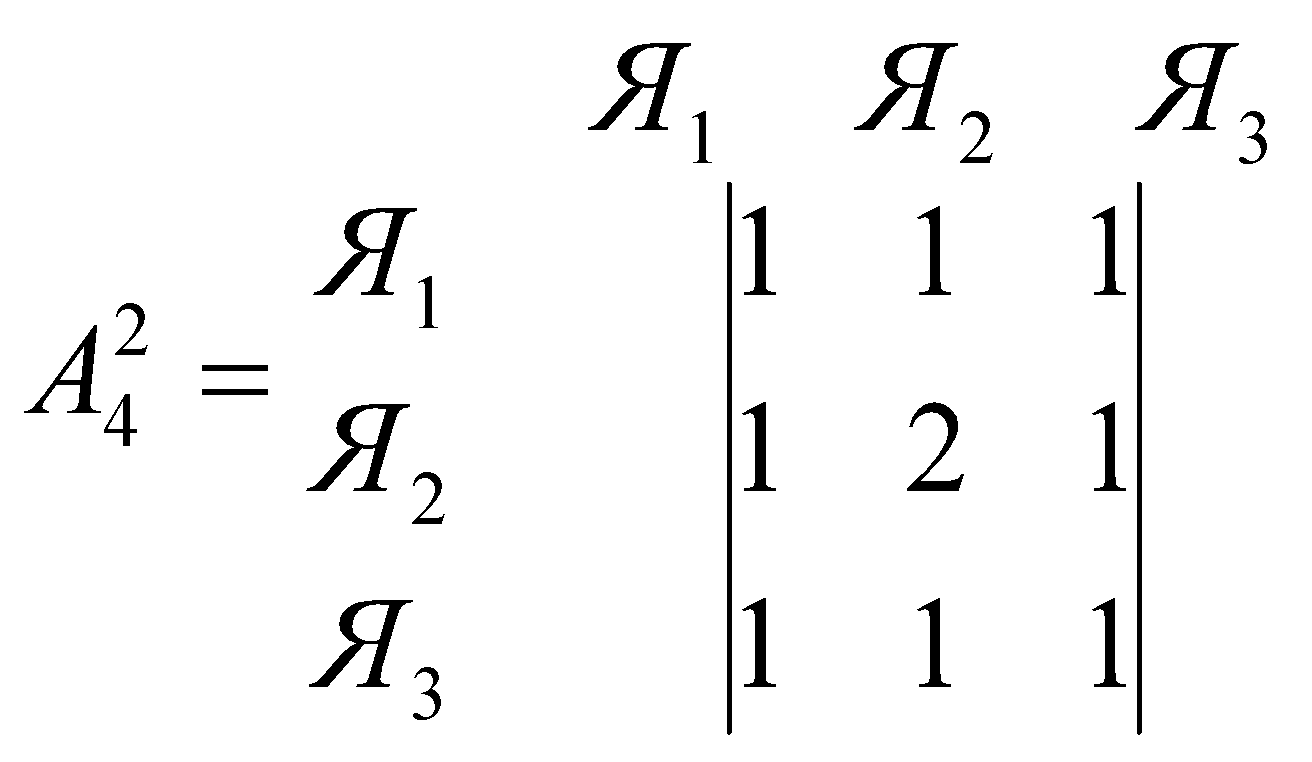


Рисунок 5 – Матрица А24 парных сравнений времени

2. Код программы:

#include <iostream>

#include <iomanip>

using namespace std;

const int nCriteria = 4;

const int nLanguage = 3;

float A1[nCriteria][nCriteria] =

{

{ 1.0, 1.0 / 5, 2, 2 },

{ 5, 6.0, 3, 3 },

{ 1.0 / 2, 1.0 / 3, 1.0, 1.0 / 6 },

{ 1.0 / 2, 1.0 / 3, 6, 1.0 }

};

float A1Copy[nCriteria][nCriteria] =

{

{ 1.0, 1.0 / 5, 2, 2 },

{ 5, 1.0, 3, 3 },

{ 1.0 / 2, 1.0 / 3, 1.0, 1.0 / 6 },

{ 1.0 / 2, 1.0 / 3, 6, 1.0 }

};

float w[nCriteria];

float wDash[nCriteria];

float lambda = 0;

float coherence = 0;

float A2[nCriteria][nLanguage \* nLanguage] =

{

{ 1.0, 2, 1.0 / 4, 1.0 / 2, 1.0, 1.0, 4, 1.0, 1.0 },

{ 1.0, 1.0, 3, 1.0, 1.0, 1.0 / 2, 1.0 / 3, 2, 1.0 },

{ 1.0, 2, 4, 1.0 / 2, 1.0, 3, 1.0 / 4, 1.0 / 3, 1.0 },

{ 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 }

};

float A2Copy[nCriteria][nLanguage \* nLanguage] =

{

{ 1.0, 2, 1.0 / 4, 1.0 / 2, 1.0, 1.0, 4, 1.0, 1.0 },

{ 1.0, 1.0, 3, 1.0, 1.0, 1.0 / 2, 1.0 / 3, 2, 1.0 },

{ 1.0, 2, 4, 1.0 / 2, 1.0, 3, 1.0 / 4, 1.0 / 3, 1.0 },

{ 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 }

};

float w2[nCriteria][nLanguage];

float w2Dash[nCriteria][nLanguage];

float lambdaLen[nLanguage] = { 0 };

float coherenceLen[nLanguage] = { 0 };

float d[nLanguage] = { 0 };

int indMax;

void normStep1(float\* matrix);

void normStep11(float\* matrix);

float normStep23(int n, float\* matrix);

float productAw(int n, float\* matrix, float\* w);

int max(int n, float\* matrix);

int main() {

normStep1(&A1Copy[0][0]);

for (int i = 0; i < nCriteria; i++) {

w[i] = normStep23(nCriteria, &A1Copy[i][0]);

}

for (int i = 0; i < nCriteria; i++) {

wDash[i] = productAw(nCriteria, &A1[i][0], &w[1]);

}

for (int i = 0; i < nCriteria; i++) {

wDash[i] /= w[i];

}

cout << endl;

for (int i = 0; i < nCriteria; i++) {

lambda += wDash[i];

}

lambda /= nCriteria;

coherence = (lambda - nCriteria) / (nCriteria - 1);

cout << "Sobstvennoe znachenie matrici A1 = " << lambda << ", IS = " << coherence << endl << endl;

for (int i = 0; i < (nCriteria); i++)

normStep11(&A2Copy[i][0]);

for (int i = 0; i < (nCriteria); i++)

for (int j = 0; j < nLanguage; j++)

w2[i][j] = normStep23(nLanguage, &A2Copy[i][j \* nLanguage]);

for (int i = 0; i < (nCriteria); i++)

for (int j = 0; j < nLanguage; j++)

w2[i][j] = productAw(nLanguage, &A2[i][j \* nLanguage], &w2[i][0]);

for (int i = 0; i < (nCriteria); i++)

for (int j = 0; j < nLanguage; j++)

w2Dash[i][j] /= w2[i][j];

for (int i = 0; i < (nCriteria); i++)

for (int j = 0; j < nLanguage; j++)

lambdaLen[i] += w2Dash[i][j];

for (int i = 0; i < (nLanguage); i++)

lambdaLen[i] /= nLanguage;

for (int i = 0; i < (nLanguage); i++)

coherenceLen[i] = (lambdaLen[i] - nLanguage) / (nLanguage - 1);

for (int i = 0; i < (nLanguage); i++)

cout << "Sobstvennoe znachenie matrici A2" << (i + 1) << " = " << lambdaLen[i] << ", IS = " << coherenceLen[i] << endl << endl;

for (int i = 0; i < nLanguage; i++)

for (int j = 0; j < nCriteria; j++)

d[i] += w[j] \* w2[i][j];

indMax = max(nLanguage, &d[0]);

cout << "The Best reshenie " << (indMax + 1) << ", s D = " << d[indMax] << endl << endl;

cout << "(c) VolK Company" << endl;

system("pause");

return 0;

}

void normStep1(float\* matrix) {

const int n = 4;

float sum[n];

int i, j;

float\* k;

for (i = 0; i < n; i++) {

sum[i] = 0;

for (j = 0; j < n; j++)

sum[i] += \*(matrix + i + j \* n);

}

for (i = 0; i < n; i++)

for (j = 0; j < n; j++) {

k = (matrix + i + j \* n);

(\*k) = (\*k) / sum[i];

}

}

void normStep11(float\* matrix) {

const int n = 3;

float sum[n];

int i, j;

float\* k;

for (i = 0; i < n - 1; i++) {

sum[i] = 0;

for (j = 0; j < n; j++)

sum[i] += \*(matrix + i + j \* n);

}

for (i = 0; i < n; i++)

for (j = 0; j < n; j++) {

k = (matrix + i + j \* n);

(\*k) = (\*k) / sum[i];

}

}

float normStep23(int n, float\* matrix) {

int i;

float w = 0;

for (i = 0; i < n; i++)

w += \*(matrix + i);

return w / n;

}

float productAw(int n, float\* matrix, float\* w) {

int i;

float wD = 0;

for (i = 0; i < n; i++)

wD += (\*(w + i)) \* (\*(matrix + i));

return wD;

}

int max(int n, float\* matrix) {

float max = \*matrix;

int index = 0;

for (int i = 0; i < n; i++)

if (max < \*(matrix + i)) {

max = \*(matrix + i);

index = i;

}

return index;

}

4 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ

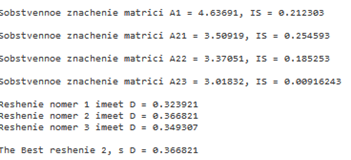


Рисунок 1 – Результаты работы программы

ВЫВОДЫ

В ходе выполнения лабораторной работы был применен метод анализа иерархий при принятии решений по выбору альтернатив. Были составлены матрицы парных сравнений характеристик. С помощью этих матриц было получено эффективное решение.