МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ОДЕССКИЙ НАЦИОНАЛЬНЫЙ МОРСКОЙ УНИВЕРСИТЕТ

Кафедра «Техническая кибернетика»

Отчет по лабораторной работе №1

Тема: «Метод анализа иерархий»

Выполнил:

студент 3к. 2гр. КСФ Талабишка Р.Р.Проверил:

Рудниченко Н.Д.

Одесса 2017

Содержание

* Введение……………………………………………………………3
* Теоретическая часть……………………………………………….4-5
* Практическая часть
* Контрольные вопросы
* Вывод

ВВЕДЕНИЕ

Цель данной лабораторной работы изучить принципы метода иерархий, произвести оценку и выбор объектов (услуг) согласно варианту выбранного индивидуального задания, используя метод анализа иерархий (МАИ).

Индивидуальное задание заключается в том, что необходимо собрать описательный материал по данной теме и привести словесное описание исследуемых вариантов своего объекта исследования. Также необходимо произвести описание, оценку и выбор наилучшего объекта из пяти вариантов по пяти критериям, согласно своему варианту, а именно средства связи, используя метод анализа иерархий.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Метод Анализа Иерархий (МАИ) – математический инструмент системного подхода к решению проблем принятия решений. МАИ не предписывает лицу, принимающему решение (ЛПР), какого-либо «правильного» решения, а позволяет ему в интерактивном режиме найти такой вариант (альтернативу), который наилучшим образом согласуется с его пониманием сути проблемы и требованиями к ее решению. Метод анализа иерархий можно применять не только для сравнения объектов, но и для решения более сложных проблем управления, прогнозирования и др.

Иерархия возникает, когда системы, функционирующие на одном уровне, функционируют как часть системы более высокого уровня, становясь подсистемами этой системы. МАИ является иерархической процедурой для иерархического представления элементов, определяющих суть проблемы. Метод состоит в декомпозиции проблемы на более простые составляющие части дальнейшей обработки последовательности суждений лица, принимающего решения по парным сравнениям. Однако МАИ включает процесс синтеза многих суждении, получения приоритетности критериев и нахождения альтернативных решений.

## МАИ включает в себя следующие этапы:

* 1. Очертить проблему и определенную цель - первый уровень иерархии.
  2. Построить иерархию, начиная с вершины:
* Первый уровень: цель
* Второй уровень: критерии
* Третий уровень: перечень альтернатив.

1. Построить множество матриц парных сравнений для каждого из нижних уровней.
2. После проведения всех парных сравнений определяются λmax и коэффициент согласованности.
3. Этапы 3, 4, 5 провести для всех уровней и групп иерархии.
4. Построить вектор глобальных приоритетов.
5. Определить результат.

В частности для определения оценки важности критериев при построении матриц парных сравнений используется таблица важности (табл. 1).

|  |
| --- |
| 1 - равная важность |
| 3 - умеренное превосходство одного над другим |
| 5 - существенное превосходство одного над другим |
| 7 - значительное превосходство одного над другим |
| 9 - очень сильное превосходство одного над другим |
| 2, 4, 6, 8 - соответствующие промежуточные значения |

Таблица 1 – Смысловая характеристика оценок

Основным **достоинством метода анализа иерархий** является высокая универсальность, но в тоже время есть необходимость получения большого объема информации от экспертов.

Практическая часть

В данной лабораторной работе, были рассмотрены средства оргтехники: сканнеры. Сканнер – устройство ввода, которое, анализируя какой-либо объект (обычно изображение, текст), создаёт его цифровое изображение. Процесс получения этой копии называется сканированием.

Следующим этапом после определения необходимых средств, заключается разработка программы с использованием метода анализа иерархий, для определения самого оптимального решения. Для расчетов используется пять вариантов с пятью критериями. Листинг реализации представлен ниже:

{

…

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String t11 = jTable1.getValueAt(0, 0).toString();

double z11 =Double.parseDouble(t11);

String t12 = jTable1.getValueAt(0, 1).toString();

double z12 =Double.parseDouble(t12);

String t13 = jTable1.getValueAt(0, 2).toString();

double z13 =Double.parseDouble(t13);

String t14 = jTable1.getValueAt(0, 3).toString();

double z14 =Double.parseDouble(t14);

String t15 = jTable1.getValueAt(0, 4).toString();

double z15 =Double.parseDouble(t15);

double ok1 = pow(z11\*z12\*z13\*z14\*z15,0.2);

String okz1 = String.format("%.5f", ok1);

jTable1.setValueAt(okz1,0, 5);

////////////////////////////////////////

String t21 = jTable1.getValueAt(1, 0).toString();

double z21 =Double.parseDouble(t21);

String t22 = jTable1.getValueAt(1, 1).toString();

double z22 =Double.parseDouble(t22);

String t23 = jTable1.getValueAt(1, 2).toString();

double z23 =Double.parseDouble(t23);

String t24 = jTable1.getValueAt(1, 3).toString();

double z24 =Double.parseDouble(t24);

String t25 = jTable1.getValueAt(1, 4).toString();

double z25 =Double.parseDouble(t25);

double ok2 = pow(z21\*z22\*z23\*z24\*z25,0.2);

String okz2 = String.format("%.5f", ok2);

jTable1.setValueAt(okz2,1, 5);

//////////////////////////////////////////

String t31 = jTable1.getValueAt(2, 0).toString();

double z31 =Double.parseDouble(t31);

String t32 = jTable1.getValueAt(2, 1).toString();

double z32 =Double.parseDouble(t32);

String t33 = jTable1.getValueAt(2, 2).toString();

double z33 =Double.parseDouble(t33);

String t34 = jTable1.getValueAt(2, 3).toString();

double z34 =Double.parseDouble(t34);

String t35 = jTable1.getValueAt(2, 4).toString();

double z35 =Double.parseDouble(t35);

double ok3 = pow(z31\*z32\*z33\*z34\*z35,0.2);

String okz3 = String.format("%.5f", ok3);

jTable1.setValueAt(okz3,2, 5);

////////////////////////////////////////////////////

String t41 = jTable1.getValueAt(3, 0).toString();

double z41 =Double.parseDouble(t41);

String t42 = jTable1.getValueAt(3, 1).toString();

double z42 =Double.parseDouble(t42);

String t43 = jTable1.getValueAt(3, 2).toString();

double z43 =Double.parseDouble(t43);

String t44 = jTable1.getValueAt(3, 3).toString();

double z44 =Double.parseDouble(t44);

String t45 = jTable1.getValueAt(3, 4).toString();

double z45 =Double.parseDouble(t45);

double ok4 = pow(z41\*z42\*z43\*z44\*z45,0.2);

String okz4 = String.format("%.5f", ok4);

jTable1.setValueAt(okz4,3, 5);

///////////////////////////////////////////////////////

String t51 = jTable1.getValueAt(4, 0).toString();

double z51 =Double.parseDouble(t51);

String t52 = jTable1.getValueAt(4, 1).toString();

double z52 =Double.parseDouble(t52);

String t53 = jTable1.getValueAt(4, 2).toString();

double z53 =Double.parseDouble(t53);

String t54 = jTable1.getValueAt(4, 3).toString();

double z54 =Double.parseDouble(t54);

String t55 = jTable1.getValueAt(4, 4).toString();

double z55 =Double.parseDouble(t55);

double ok5 = pow(z51\*z52\*z53\*z54\*z55,0.2);

String okz5 = String.format("%.5f", ok5);

jTable1.setValueAt(okz5,4, 5);

////////////////////////////////////////////

double sm1 = z11+z21+z31+z41+z51;

String sum1 = Double.toString(sm1);

jTextField1.setText(sum1);

jTextField1.setEditable(false);

double sm2 = z12+z22+z32+z42+z52;

String sum2 = Double.toString(sm2);

jTextField2.setText(sum2);

jTextField2.setEditable(false);

double sm3 = z13+z23+z33+z43+z53;

String sum3 = Double.toString(sm3);

jTextField3.setText(sum3);

jTextField3.setEditable(false);

double sm4 = z14+z24+z34+z44+z54;

String sum4 = Double.toString(sm4);

jTextField4.setText(sum4);

jTextField4.setEditable(false);

double sm5 = z15+z25+z35+z45+z55;

String sum5 = Double.toString(sm5);

jTextField5.setText(sum5);

jTextField5.setEditable(false);

////////////////////////////////////////////

double smo = ok1+ok2+ok3+ok4+ok5;

String sumo = String.format("%.5f", smo);

jTextField6.setText(sumo);

jTextField6.setEditable(false);

///////////////////////////////////////////

double no1 = ok1/smo;

String nov1 = String.format("%.5f", no1);

jTable1.setValueAt(nov1,0, 6);

double no2 = ok2/smo;

String nov2 = String.format("%.5f", no2);

jTable1.setValueAt(nov2,1, 6);

double no3 = ok3/smo;

String nov3 = String.format("%.5f", no3);

jTable1.setValueAt(nov3,2, 6);

double no4 = ok4/smo;

String nov4 = String.format("%.5f", no4);

jTable1.setValueAt(nov4,3, 6);

double no5 = ok5/smo;

String nov5 = String.format("%.5f", no5);

jTable1.setValueAt(nov5,4, 6);

//////////////////////////////////////////////

double pr1 = no1\*sm1;

String pro1 = String.format("%.5f", pr1);

jTextField7.setText(pro1);

jTextField7.setEditable(false);

double pr2 = no2\*sm2;

String pro2 = String.format("%.5f", pr2);

jTextField8.setText(pro2);

jTextField8.setEditable(false);

double pr3 = no3\*sm3;

String pro3 = String.format("%.5f", pr3);

jTextField9.setText(pro3);

jTextField9.setEditable(false);

double pr4 = no4\*sm4;

String pro4 = String.format("%.5f", pr4);

jTextField10.setText(pro4);

jTextField10.setEditable(false);

double pr5 = no5\*sm5;

String pro5 = String.format("%.5f", pr5);

jTextField11.setText(pro5);

jTextField11.setEditable(false);

////////////////////////////////////////////

double lm = pr1+pr2+pr3+pr4+pr5;

String lmax = String.format("%.5f", lm);

jTextField12.setText(lmax);

jTextField12.setEditable(false);

/////////////////////////////////////////////

double is = (lm-n)/(n-1);

String IS = String.format("%.5f", is);

jTextField13.setText(IS);

jTextField13.setEditable(false);

/////////////////////////////////////////////

double os = is/k;

String OS = String.format("%.5f", os);

//String OS = Double.toString(os);

jTextField14.setText(OS);

jTextField14.setEditable(false);

/////////////////////////////////////////////

double [] array = { no1, no2, no3, no4, no5};

double x = array[0];

for(int i = 0; i<array.length; i++){

if(x<array[i])

x = array[i];

}

String X = String.format("%.5f", x);

jTextField15.setText(X);

jTextField15.setEditable(false); }

Таким образом, мы получили форму для ввода данных и их дальнейшего расчета. Результат выполнения программы представлен на рисунках 1 и 2 .

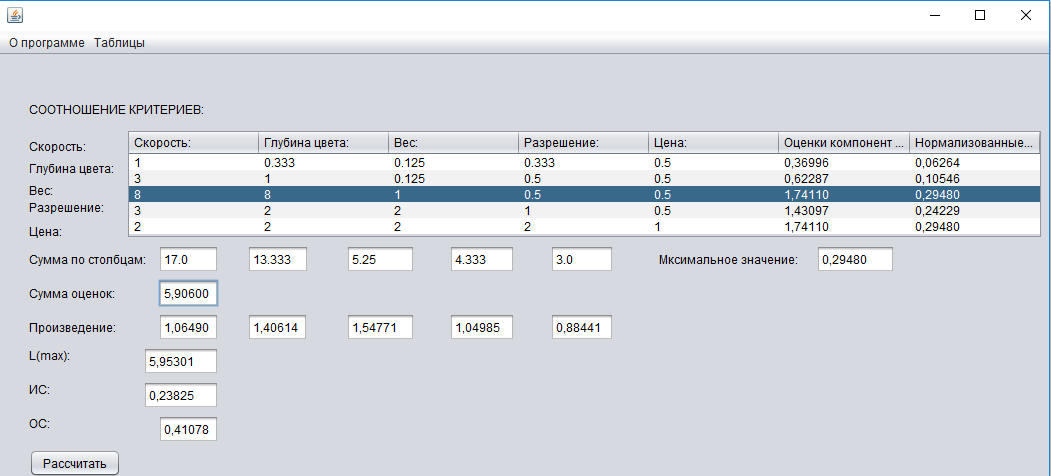


Рисунок 1 – Окно ввода значений и вывода результатов МАИ

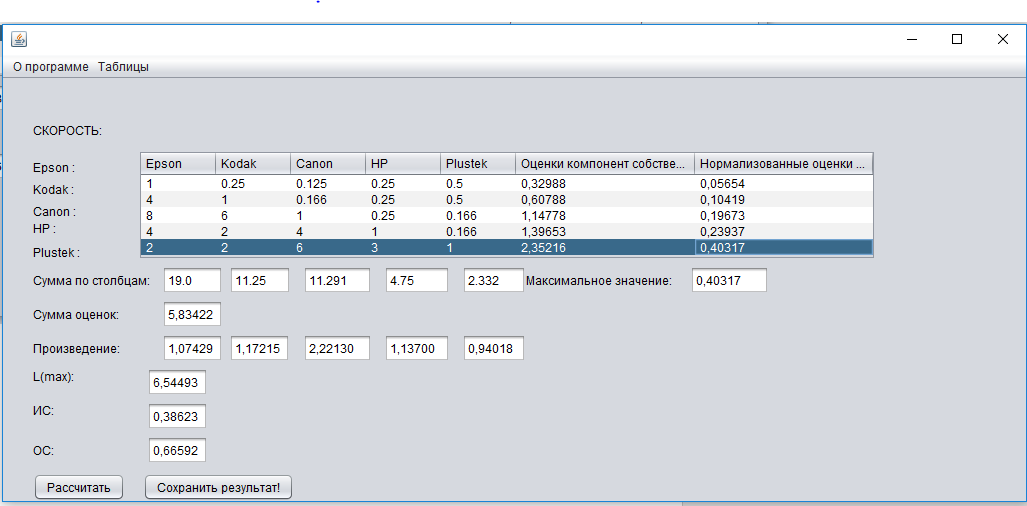


Рисунок 2 – Окно ввода приоритетов по критерию «Скорость сканирования»

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите основные этапы метода анализа иерархий.

## К основным этапам метода анализа иерархий можно отнести:

1. Очертить проблему и определенную цель - первый уровень иерархии.
2. Построить иерархию, начиная с вершины:
3. Первый уровень: цель
4. Второй уровень: критерии
5. Третий уровень: перечень альтернатив.
6. Построить множество матриц парных сравнений для каждого из нижних уровней.
7. После проведения всех парных сравнений определяются λmax и коэффициент согласованности.
8. Этапы 3, 4, 5 провести для всех уровней и групп иерархии.
9. Построить вектор глобальных приоритетов.
10. Определить результат.
11. Опишите процесс попарного сравнения объекта по какому-либо признаку.

Попарное сравнение критериев по важности по девятибалльной шкале с составлением соответствующей матрицы (таблицы) размера (n х n). Система парных сведений приводит к результату, который может быть представлен в виде обратно симметричной матрицы. Элементом матрицы a(i,j) является интенсивность проявления элемента иерархии i относительно элемента иерархии j, оцениваемая по шкале интенсивности от 1 до 9, где оценки имеют следующий смысл:

* равная важность – 1;
* умеренное превосходство – 3;
* значительное превосходство – 5;
* сильное превосходство – 7;
* очень сильное превосходство – 9;
* в промежуточных случаях ставятся четные оценки: 2, 4, 6, 8 (например, 4 – между умеренным и значительным превосходством).

При этом при проведении попарных сравнений в основном ставятся следующие вопросы при сравнении элементов А и Б:

* какой из них важнее или имеет большее воздействие?
* какой из них более вероятен?
* какой из них предпочтительнее?

1. Опишите шкалу выбора приоритетов.

Для проведения парных сравнений задаются параметры: шкала сравнений и способ сравнений. При проведении парного сравнения объектов  и  достаточно установить только один из результатов  (оценка отношения «веса» объекта  и весу объекта) или, так как.Шкала сравнений – упорядоченный набор градаций (терминов, чисел и т.п.) для выражения результатов парных сравнений. Шкала сравнений позволяет выражать оценки отношений значений приоритетов узлов, поэтому ее деления – безразмерные величины. Шкала является количественной, если результаты парных сравнений выражаются непосредственно с помощью чисел. Шкала является качественной, если результаты парных сравнений выражаются с помощью с градаций-предпочтений. Градациям качественных шкал, использующихся в методе анализа иерархий, соответствуют числа. Т.е. качественные шкалы предоставляют возможность опосредованного оценивания приоритетов через предпочтения.

1. Перечислите основные свойства матрицы попарных сравнений.

**Способ сравнений** определяется набором парных сравнений, необходимых для определения приоритетов узлов кластера. **Матрица сравнений** – таблица числовых значений парных сравнений (для узлов кластера или для кластеров, имеющих общую вершину).

1. Как происходит формирование вектора локальных приоритетов?

**Вектор приоритетов кластера** – рейтинг узлов кластера. Вектор приоритетов узлов кластера может задаваться напрямую (без проведения сравнений) или рассчитываться на основе матрицы сравнений. Каждой альтернативе (каждому возможному решению) ставится в соответствие положительное число – приоритет. Приоритет количественно выражает важность (предпочтительность, вероятность, оптимальность и т.п.) альтернативы в соответствии с главным критерием. Сумма приоритетов всех альтернатив равна единице. Вследствие этого часто допустимо отождествление приоритетов с вероятностями. Для поддержки принятия решения в основном с помощью итогового вектора приоритетов производится интерпретация результатов применения метода. После чего можно составить сводную матрицу локальных приоритетов путем последовательной записи векторов – столбцов локальных приоритетов.

1. Опишите процесс свертки сводной матрицы локальных приоритетов.

На основе матриц попарных сравнений мы получаем векторы локальных приоритетов по каждому рассматриваемому критерию оценки. Для этого необходимо произвести свертку каждой матрицы попарных сравнений в вектор, затем любым из известных способов нормировать полученные векторы и перемножить матрицы попарных сравнений на соответствующие им нормированные векторы. Процесс свертки заключается в том, что сначала необходимо определить оценки компонент собственного вектора, а затем вычислить нормализованные оценки вектора приоритета для каждого критерия, разделив значение оценки собственного вектора на его сумму.

1. На основании чего происходит выбор оптимального варианта в методе анализа иерархий?

Выбор оптимального варианта в методе анализа иерархий происходит при помощи определения максимального значения вектора приоритета.

1. Используются ли в методе анализа иерархий основные принципы синтеза сложных систем.
2. Можно ли отнести метод анализа иерархий к методам экспертных оценок?

Экспертное оценивание — это процедура получения оценки проблемы на основе мнения специалистов (экспертов) с целью последующего принятия решения (выбора). Экспертный подход позволяет решать задачи, не поддающиеся решению обычным аналитическим способом, в том числе при помощи выбор лучшего варианта решения среди имеющихся.

1. Опишите процесс получения вектора глобальных приоритетов.

Подсчет значения глобального приоритета для каждой из альтернатив происходит как сумма произведений значения вектора приоритета для критерия и значения вектора локального приоритета этой альтернативы в отношении данного критерия.

Вывод

В ходе работы программы, мы получили оптимальный вариант решения выбора сканера. Изучили метод анализа иерархий и закрепили практическим примером.