Федеральное агентство по рыболовству

Федеральное государственное образовательное учреждение

Высшего профессионального образования

Мурманский государственный технический университет

Кафедра вычислительной математики и программного обеспечения ЭВМ

Расчетно-графическое задание

по дисциплине

Теория принятия решений

Тема:

Метод последовательных сравнений

Выполнил: студент ПТФ, группы П-561

Воронкова А.В.

Проверил: Зав. кафедры ВМ и ПО ЭВМ

Середа В.И.

Мурманск 2010

Оглавление

1. Цель работы

[2. Теоретические сведения](#_Toc278724399)

2.1 Экспертные методы в исследовании систем управления

[2.2 Метод последовательных сравнений](#_Toc278724412)

2.3 Проблемы, для решения которых привлекаются методы экспертных оценок

[3. Описание интерфейса разработанного программного продукта](#_Toc278724427)

4. Листинг

[4.1 Класс «CombinationGenerator»](#_Toc278724435)

4.2 Класс «ConditionPanel»

[4.3 Класс «MainFrame»](#_Toc278724437)

Выводы

[Список использованной литературы](#_Toc278724449)

Список использованных программных средств

[Приложение](#_Toc278724451)

## Цель работы

Программно реализовать интерактивный метод последовательных сравнений

# 2.Теоретические сведения

## Экспертные методы в исследовании систем управления

Экспертные методы вначале использовались в основном в области науки и техники, а затем начали широко применяться в управлении и экономике. Сущность этих методов как при решении задач ИСУ, так и при использовании их в практике обобщённого мнения (суждения) специалистов-экспертов по рассматриваемым вопросам. Это обобщённое мнение получается в результате усреднения различными способами мнений специалистов-экспертов.

Методы экспертных оценок – это научные методы анализа сложных проблем. Эксперты проводят интуитивно-логический анализ проблемы с количественной оценкой суждений, с формальной обработкой результатов. Их обобщённое мнение, полученное в результате обработки индивидуальных оценок, принимается как решение проблемы.

Эти методы предполагают организацию специальной процедуры получения информации, когда специалисты в области решаемой проблемы (эксперты) используют количественные методы, как при организации процедуры экспертной оценки, так и при обработке её результатов.

Исследуемые или оцениваемые с помощью методов экспертных оценок объекты или явления различаются на основе признаков, характеристик, параметров, показателей. Как правило, каждый показатель отражает некоторое свойство, характеристику объекта.

В общем случае это свойство может быть измерено, отображено несколькими способами. В тоже время, какой бы из способов мы ни избрали, должны сохраняться неизменными некоторые соотношения значений показателей для различных объектов.

С каждым показателем связывается с одной стороны некоторая характеристика, остающаяся постоянной при любых допустимых способах его измерения. А с другой стороны множество преобразований перехода от допустимого способа измерения данного показателя к другому.

Если два показателя имеют совпадающие множества допустимых преобразований, то говорят, что показатели имеют шкалу одного типа.

Если допустимо преобразование: умножение на положительную константу, то соответствующая шкала называется «шкалой отношений».

Если к этому преобразованию можно дополнить следующее преобразование: добавить константу, такая шкала называется «шкалой интервалов».

Показатели, имеющие шкалу не менее совершенную, сем шкала интервалов, называются количественными. Могут встретиться показатели с т.н. номинальной шкалой. В этом случае устанавливаются только тождества или различия. Показатели с порядковыми шкалами принято называть качественными.

При классификации по признаку оценки предпочтений при принятии решений в настоящее время наиболее распространены метод рангов, метод непосредственного оценивания, метод сопоставлений. Он включает в себя две разновидности: парного сравнения и последовательного сравнения.

## 2.2 Метод последовательных сравнений

Общим недостатком показателей, получаемых на основе суммирования баллов, является то, что недостаток качества по одному из них можно компенсировать за счет других, получая один и тот же результат при различной значимости факторов. Поэтому для повышения надежности оценок, значение имеет выявление связей и установление зависимостей между всеми значимыми факторами. Установление таких зависимостей возможно методом последовательных сравнений (У. Черчмен, Р. Акоф).

Процедура состоит в следующем. Эксперту предоставляется перечень факторов (критериев, альтернатив, результатов), которые необходимо оценить по их относительной важности и он производит ранжирование. Наиболее важному фактору приписывается оценка (вес ) = 1, а остальным оценки () между 0 и 1 в порядке относительной важности.

Затем эксперт устанавливает, является фактор с оценкой 1 более важным, чем комбинация остальных факторов.

* Если да, то он увеличивает оценку , чтобы она была больше, чем сумма всех остальных, т.е.



* Если нет, то он корректирует оценку  (если необходимо), чтобы она была меньше суммы всех остальных, т.е.



Далее определяется, является ли второй фактор с оценкой  более важным, чем все остальные. И так далее до  фактора.

Таким образом, процедура состоит в систематической проверке оценок на базе их последовательного сравнения.

Общая процедура метода следующая:

1. Упорядочить результаты в соответствии с их значимостью (относительной важностью) с точки зрения эксперта.

Пусть  представляет наиболее важный результат,  – следующий по степени важности и т. д., а – наименее важный.

1. Приписать вес 1,00 результату  (т.е.  = 1,00) и другие веса всем остальным результатам.
2. Сравнить

 с  +  + ... + :

* + если  предпочтительнее, надо изменить (в случае необходимости) значение  так, чтобы  >  +  + … + . При этой корректировке, так же как и при всех остальных, следует стремиться к тому, чтобы веса набора (,  и т.д.) остались без изменений. Далее следует перейти к шагу 4.
	+ если  =  +  + ... + , то изменить (в случае необходимости) значение , так чтобы выполнялось равенство  =  +  + … + и затем перейти к шагу 4
	+ если результат  менее предпочтителен, чем  +  + ... + , то изменить значение  так, чтобы выполнялось неравенство  <  +  + … + . Далее сравнить  с  +  + ... +  и повторять до тех пор, пока  будет или предпочтительнее, или равноценен всем остальным результатам.
1. Сравнить  с  + ... + и выполнить весь шаг 3.
2. Продолжить шаг 4 до тех пор, пока не будет выполнено сравнение с + .
3. Преобразовать каждое полученное значение  в нормированное ′, разделив соответствующие веса на . В итоге ′ должна быть равна 1,00.

Если число результатов больше семи, метод становится громоздким. В этом случае применяется процедура разбиения на подмножества, к каждому из которых применяется эта процедура.

## экспертный последовательный интерактивный сравнение

## 2.3 Проблемы, для решения которых привлекаются методы экспертных оценок

Экспертные методы вначале использовались в основном в области науки и техники, а затем начали широко применяться в управлении и экономике. Сущность этих методов как при решении задач ИСУ, так и при использовании их в практике обобщённого мнения (суждения) специалистов-экспертов по рассматриваемым вопросам. Это обобщённое мнение получается в результате усреднения различными способами мнений специалистов-экспертов.

Это научные методы анализа сложных проблем. Эксперты проводят интуитивно-логический анализ проблемы с количественной оценкой суждений, с формальной обработкой результатов. Их обобщённое мнение, полученное в результате обработки индивидуальных оценок, принимается как решение проблемы.

Проблемы, для решения которых приходится применять методы экспертных оценок, часто делят на два класса, каждый из которых определённым образом влияет на этапы и процедуры проведения экспертизы.

Проблемы первого класса характеризуются тем, что в их отношении в целом имеется достаточная информация, но она может носить качественный характер или имеется многокритериальность, что вызывает необходимость привлечения экспертов. Основные задачи, которые в данном случае приходится решать при использовании методов экспертных оценок, состоит в поиске хороших экспертов и правильной организации процедуры экспертизы, при этом полагают, что групповое мнение экспертов приближается к истинному значению оцениваемых параметров. В этом случае при обработке оценок широко используются методы математической статистики. Эксперты чаще всего используют порядковую и интервальную шкалы.

Проблемы, составляющие второй класс, не характеризуются достаточным информационным потенциалом. К ним чаще относятся проблемы, возникающие при решении большинства задач прогнозирования. Здесь эксперты используют чаще номинальную и порядковую шкалы.

К разновидностям экспертного метода можно отнести социологический анализ, который основан на опросе, сборе и анализе мнений респондентов. Например, фактических или потенциальных потребителей. Такой опрос и сбор мнений производится обычно в письменной форме (анкеты), либо устно (конференции, аукционы, выставки).

При использовании этого метода также следует применять научно-обоснованные способы опроса, математические принципы сбора и обработки информации. Обработка экспертных и социологических данных, расчёты мер согласованности требуют трудоёмких вычислений. Поэтому при сборе и обработке результатов экспертной и социологической информации используют вычислительную технику и специально разработанные программные продукты.

# Описание интерфейса разработанного программного продукта

Окно приложения подразделяется на две основные области: область ввода данных для реализации Метода последовательных сравнений и область анализа и выполнения указанного метода.

Область ввода данных для реализации метода

Область анализа и выполнения метода

Длина первой области составляет 350 px, длина второй области – 230 px. Соотношение длин областей составляет 65%, в свою очередь, золотое сечение представляет отношение 2:3. Таким образом, созданный интерфейс программного продукта полностью удовлетворяет требованиям пропорциональности.

Предложения по улучшению интерфейса: проанализировав пропорции элементов управления относительно друг друга и края формы, а также пропорции основных областей формы, можно сделать вывод о корректности построения спроектированного интерфейса.

Альтернативным является вариант организации интерфейса, когда основные области: область ввода данных для реализации метода и область анализа расположены на разных вкладках. Однако предложенный альтернативный вариант не является оптимальным с точки зрения минимизации временных интервалов, так как при таком расположении основных областей, пользователь будет вынужден совершать лишние перемещения мыши между указанными областями и лишние клики по рабочей области.

Таким образом, реорганизация анализируемого интерфейса не целесообразна. Спроектированный интерфейс является оптимальным, лаконичным и простым в использовании.

# Листинг

##

## 4.1 Класс «CombinationGenerator»

public class CombinationGenerator {

private int[] a;

private int n;

private int r;

private BigInteger numLeft;

private BigInteger total;

public CombinationGenerator(int n, int r) {

if (r > n) {

throw new IllegalArgumentException();

}

if (n < 1) {

throw new IllegalArgumentException();

}

this.n = n;

this.r = r;

a = new int[r];

BigInteger nFact = getFactorial(n);

BigInteger rFact = getFactorial(r);

BigInteger nminusrFact = getFactorial(n - r);

total = nFact.divide(rFact.multiply(nminusrFact));

reset();

}

//------

// Reset

//------

public void reset() {

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

a[i] = i;

}

numLeft = new BigInteger(total.toString());

} // Return number of combinations not yet generated

//------------------------------

public BigInteger getNumLeft() {

return numLeft;

}

//-----------------------------

// Are there more combinations?

//-----------------------------

public boolean hasMore() {

return numLeft.compareTo(BigInteger.ZERO) == 1;

} // Return total number of combinations

//--------------------------

public BigInteger getTotal() {

return total;

}

//------------------

// Compute factorial

//------------------

private static BigInteger getFactorial(int n) {

BigInteger fact = BigInteger.ONE;

for (int i = n; i > 1; i--) {

fact = fact.multiply(new BigInteger(Integer.toString(i)));

}

return fact;

}

//--------------------------------

// Generate next combination (algorithm from Rosen p. 286)

//----------------------------------

public int[] getNext() {

if (numLeft.equals(total)) {

numLeft = numLeft.subtract(BigInteger.ONE);

return a;

}

int i = r - 1;

while (a[i] == n - r + i) {

i--;

}

a[i] = a[i] + 1;

for (int j = i + 1; j < r; j++) {

a[j] = a[i] + j - i;

}

numLeft = numLeft.subtract(BigInteger.ONE);

return a;

}

}

## 4.2 Класс «ConditionPanel»

public class ConditionPanel extends javax.swing.JPanel {

public ConditionPanel() {

initComponents();

}

public void setCondition(int indices[], int num) {

String s;

s = " " + (indices[0] + 1) + " > ";

for (int i = 1; i < indices.length - 1; i++) {

s += (indices[i] + 1) + " + ";

}

s += (indices[indices.length - 1] + 1);

jLabel1.setText("" + num + ". " + s);

}

public void check() {

jCheckBox1.setSelected(!jCheckBox1.isSelected());

}

boolean isSelected() {

return jCheckBox1.isSelected();

}

}

## 4.3 Класс «MainFrame»

public class MainFrame extends javax.swing.JFrame {

/\*\* Creates new form MainFrame \*/

public MainFrame() {

initComponents();

jList1.setCellRenderer(new ListCellRenderer() {

public Component getListCellRendererComponent(JList list, Object value, int index, boolean isSelected, boolean cellHasFocus) {

Component comp = (Component)value;

comp.setBackground(isSelected ? Color.LIGHT\_GRAY : Color.white);

return comp;

}

});

jList1.addMouseListener(new MouseAdapter() {

@Override

public void mouseReleased(MouseEvent e) {

vec.get(jList1.getSelectedIndex()).check();

jList1.updateUI();super.mouseReleased(e);

}

});

jSpinner1.setValue(2);

}

private Vector<ConditionPanel> vec;

private LinkedList<int[]> conditions;

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

int value = (Integer)jSpinner1.getValue();

conditions = new LinkedList<int[]>();

int[] lastNumIndex = new int[value];

for(int num : lastNumIndex) {

System.out.println(num);

}

for (int i = 3; i <= value; i++) {

int[] indices;

CombinationGenerator x = new CombinationGenerator(value, i);

while (x.hasMore()) {

indices = x.getNext();

conditions.add(indices.clone());

}

}

Collections.sort(conditions, new Comparator<int[]>() {

public int compare(int[] o1, int[] o2) {

if( o1[0] < o2[0] ) {

return -1;

} else if( o1[0] == o2[0] ) {

if(o1.length < o2.length) {

return -1;

} else {

return 0;

}

} else {

return 1;

}

}

});

vec = new Vector<ConditionPanel>();

int num = 1;

for(int[] indices : conditions) {

ConditionPanel list = new ConditionPanel();

list.setCondition(indices, num);

num++;

vec.add(list);

}

jList1.setListData(vec);

jList1.updateUI();

}

private void jSpinner1StateChanged(javax.swing.event.ChangeEvent evt) {

jTable1.setModel(new DefaultTableModel((Integer)jSpinner1.getValue(), 3)

boolean[] canEdit = new boolean [] {

false, true, true

};

@Override

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

@Override

public String getColumnName(int column) {

String name = "";

switch(column) {

case 0:

name = "№";

break;

case 1:

name = "цель";

break;

case 2:

name = "оценка";

break;

}

return name;

}

});

for (int i = 0; i < (Integer)jSpinner1.getValue(); i++){

String s = "" + (1-0.1\*i);

jTable1.setValueAt(i + 1, i, 0);

jTable1.setValueAt("цель " + (i + 1), i, 1);

jTable1.setValueAt(s, i, 2);

}

}

private float[] values;

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

jTable1.updateUI();

int count = (Integer)jSpinner1.getValue();

values = new float[count];

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

values[i] = Float.parseFloat((String) jTable1.getModel().getValueAt(i, 2));

}

if (jRadioButton1.isSelected()) {

manualSolve();

} else {

automaticSolve();

}

}

}

private void manualSolve() {

int i = checkConditions();

if (i !=-1){

JOptionPane.showMessageDialog(this, (i + 1) + " правило не выполнилось. Скорректируйте оценки");

}

else{

norm();

JOptionPane.showMessageDialog(this, "Принятые оценки корректны");

}

}

private void automaticSolve() {

ArrayList<Integer> ind = new ArrayList<Integer>();

ind.add(0);

for (int i = 1; i < conditions.size(); i++) {

int[] currentIndex = conditions.get(i);

int[] previousIndex = conditions.get(i - 1);

if (currentIndex[0] == previousIndex[0]

&& currentIndex.length == previousIndex.length) {

if (isConditionSelected(i) != isConditionSelected(i - 1)) {

JOptionPane.showMessageDialog(this, "Уловия не могут быть выполнены при данных значениях оценок !!!");

return;

}

}

if (currentIndex[0] == previousIndex[0]

&& currentIndex.length > previousIndex.length) {

if (isConditionSelected(i) == true && isConditionSelected(i - 1) == false) {

JOptionPane.showMessageDialog(this, "Уловия не могут быть выполнены при данных значениях оценок !!!");

return;

}

}

if(currentIndex.length != previousIndex.length) {

ind.add(i);

}

}

{

int first = 0;

ArrayList<Integer> sortedInd = new ArrayList<Integer>();

for (int size = values.length - 2; size > 0 ; size--) {

List<Integer> sublist = ind.subList(first, first + size);

Collections.reverse(sublist);

sortedInd.addAll(sublist);

first += size;

}

ind = sortedInd;

}

int n = ind.size();

float delta = 0.01f;

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

int conditionNum = ind.get(i);

int[] cond = conditions.get(conditionNum);

boolean solving = calculute(conditionNum, values);

boolean selected = isConditionSelected(conditionNum);

if (solving != selected) {

if (solving) {

float value = getSumm(conditionNum, values) - delta;

if (value > values[cond[0] + 1]) {

values[cond[0]] = value;

} else {

JOptionPane.showMessageDialog(this, "Уловия не могут быть выполнены при данных значениях оценок");

return;

}

} else {

float value = getSumm(conditionNum, values) + delta;

if(cond[0]==0){

values[cond[0]] = value;

} else {

float d = value - values[cond[0]];

values[cond[0]] = value;

for(int j = 0; j < cond[0]; j++) {

values[j] += d;

}

}

}

}

}

norm();

JOptionPane.showMessageDialog(this, "Принятые оценки корректны");

}

private int checkConditions() {

return checkConditions(conditions.size()-1, values);

}

private boolean isConditionSelected(int numCondition){

return vec.get(numCondition).isSelected();

}

private int checkConditions(int lastCondition, float[] values) {

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

boolean selected = isConditionSelected(i);

if (calculute(i, values) == !selected) {

return i;

}

}

return -1;

}

private boolean calculute(int i, float[] values) {

int[] indices = conditions.get(i);

float leftValue = values[indices[0]];

float summ = 0;

for (int j = 1; j < indices.length; j++) {

summ += values[indices[j]];

}

return leftValue > summ;

}

private float getSumm(int i, float[] values) {

int[] indices = conditions.get(i);

float summ = 0;

for (int j = 1; j < indices.length; j++) {

summ += values[indices[j]];

}

return summ;

}

public void norm() {

int count = values.length;

float val = values[0];

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

values[i] = values[i] / val;

jTable1.setValueAt("" + values[i], i, 2);

}

}

}

# Выводы

Основное преимущество методов экспертных оценок – возможность их применения в условиях повышенного риска и неопределенности. Эта неопределенность чаще всего является следствием вероятного характера исследуемых явлений, невозможности точного предсказания окончательных исходов многих процессов и т.д. Привлечение экспертов для принятия решений позволяет снизить уровень неопределенности и повысить достоверность решений. В общем случае предполагается, что мнение группы экспертов надежнее, чем мнение отдельного индивидуума. Главное преимущество групповой оценки заключается в уменьшении различий во мнениях, в возможности получения в какой-то степени обобщенного и более представительного мнения.

Специфика и разнообразие решаемых при участии экспертов проблем существенно ограничивает возможности создания единых универсальных правил и моделей экспертизы. Однако можно ориентировочно наметить следующие основные этапы проведения экспертизы:

* формулирование цели экспертизы и разработка процедуры опроса;
* формирование группы специалистов-аналитиков;
* отбор и формирование группы экспертов;
* проведение опроса;
* анализ и обработка информации, полученной от экспертов;
* синтез объективной (статистической) информации и информации, полученной от экспертов, с целью приведения их в форму, удобную для принятия решения.

Последовательность и содержание этих этапов будут изменяться в зависимости от реальных условий и ограничений при проведении экспертизы.

Математико-статистические методы экспертных оценок получают все более широкое применение в практике принятия решений. Однако существует ряд проблем и задач, требующих дальнейших теоретических исследований и практической проверки. Можно указать на необходимость совершенствования системы отбора экспертов, повышения надежности характеристик группового мнения, разработки методов проверки обоснованности оценок и ряд других.

# Список использованной литературы

1. Электронный ресурс: Математико-статистические методы в менеджменте, http://window.edu.ru/window\_catalog/pdf2txt?p\_id=3216&p\_page=2
2. Электронный ресурс: Исследование Систем Управления, <http://www.uproizvod.ru/index.php?option=com_content&task=view&id=7&Itemid=34&limit=1&limitstart=0>

# Список использованных программных средств

1. **NetBeans IDE**, version number 6.8

# Приложение 1

