Черт-отчет для Василича

Оглавление

[Введение 3](#_Toc122097885)

[Практическая и математическая постановка задачи 4](#_Toc122097886)

[Анализ существующих алгоритмов решения 5](#_Toc122097887)

[Анализ аксиоматического метода 5](#_Toc122097888)

[Анализ прямых методов 6](#_Toc122097889)

[Анализ метода компенсации 7](#_Toc122097890)

[Анализ метода анализа иерархий 8](#_Toc122097891)

[Описание разрабатываемого алгоритма 10](#_Toc122097892)

[Укрупненная схема 10](#_Toc122097893)

[Развернутая схема 10](#_Toc122097894)

[Решение контрольного примера 13](#_Toc122097895)

[Листинг 15](#_Toc122097896)

[Результат машинного решения 18](#_Toc122097897)

[Заключение 21](#_Toc122097898)

[Список литературы 22](#_Toc122097899)

# Введение

Проблема выбора заключается в определении различных альтернатив действий и выборе оптимальной альтернативы, т. е. такой, которая позволяет получить наилучший результат в достижении поставленной цели.

В качестве альтернатив могут выступать новые целевые области (товарные рынки), виды выпускаемой продукции, инвестиции в различные сферы деятельности предприятия и т. д. Как правило, они не могут быть реализованы одновременно.

Целенаправленный выбор среди подобных альтернатив представляет собой принятие такого решения, которое предоставило бы как можно больше преимуществ.

Реализация (осуществление) любой возможной альтернативы ведет к одному или нескольким последствиям (результатам). Ожидаемыми результатами могут быть выручка от реализации товаров, издержки производства, доля удовлетворения спроса, прибыль, затраты на продвижение товара, доля рынка и др.

Таким образом, актуальность темы обусловлена необходимостью разработки программы, которая предоставляет определенные альтернативы и показывает, какая из них является оптимальной для лица, принимающего решения.

# Практическая и математическая постановка задачи

Задача выбора состоит в том, что необходимо принять такое решение, которое гарантировало бы предприятию не только прибыль, но и избавило бы от убытков. Таких задач может быть множество в зависимости от компании, например: в аэропорту необходимо составить оптимальное расписание, в благотворительном фонде распределить денежные средства и т.д. Данные задачи решает такая область математического анализа как критериальный анализ.

Задача критериального анализа состоит в выборе лучшей альтернативы среди нескольких предложенных по определенным критериям. Критериев может быть несколько, также они могут иметь разные единицы измерения, а показатели непостоянны.

Одним из методов теории принятия решений является метод анализа иерархий. Метод анализа иерархий создан американским ученым Т. Саати. Общая структура метода анализа иерархий может включать несколько иерархических уровней со своими критериями.

Метод состоит из совокупности следующих этапов:

1. Первый этап заключается в структуризации задачи в виде иерархической структуры с несколькими уровнями.

Далее будут рассмотрены методы решения задачи выбора в условиях неопределенности.

# Анализ существующих алгоритмов решения

Неопределенность – это состояние ограниченного знания текущих условий или будущих результатов. Это важный компонент риска, который связан с вероятностью и масштабом негативных последствий. Управление неопределенностью и риском также включает смягчение или даже удаление вещей, которые препятствуют эффективному принятию решений или неблагоприятно влияют на производительность.

Причинами неопределенности могут являться вещи, которые должны вскоре произойти, легче оценить, чем те, которые произойдут в дальнейшем будущем. Один из подходов к решению проблемы неопределенности заключается в том, чтобы отложить принятие решений, пока данные не станут более доступными и надежными. Конечно, отсрочка некоторых решений может привести к собственному набору рисков, особенно когда потенциальные негативные последствия ожидания велики.

## Анализ аксиоматического метода

Аксиоматический метод, в логике, процедура, посредством которой вся целая система (например, наука) генерируется в соответствии с указанными правилами путем логического вывода из некоторых основных положений (аксиом или постулатов), которые, в свою очередь, строятся из нескольких терминов, принятых в качестве начальных данных. Эти термины и аксиомы могут быть либо произвольно определены и построены, либо задуманы в соответствии с моделью, в которой, как считается, существует какой-то интуитивный критерий, указывающий на их истинность. Самыми старыми примерами аксиоматизированных систем являются силлогистика Аристотеля и геометрия Евклида.

Все методы оценки критериальных альтернатив, так или иначе, используют измерение полезности, но аксиоматические методы подходят к этим измерениям наиболее теоретически: они рассматривают их как определенные шаги, подтверждающие справедливость выбора некоторых аксиом и ведущие к возможности использования определенной функции полезности.

Аксиоматические методы непосредственно опираются на теорию ожидаемой полезности фон Неймана и Моргенштерна либо на более современные критериальной теории полезности.

Следует отметить искусственность аксиоматических методов. Главная практическая проблема, – многокритериальное сравнение альтернатив, – отступает на второй план перед чисто формальной (математической) проблемой поиска функции полезности в определенной форме.

## Анализ прямых методов

Здесь зависимость общей полезности альтернативы от оценок по отдельным критериям считается известной заранее.

Существует большое число методов, в которых форма зависимости результирующей полезности альтернативы от ее оценок по многим критериям задается без всяких теоретических оснований, а параметры этой зависимости задаются так:

* нормативно, на основании утверждённых методик («закон есть закон»);
* непосредственно устанавливаются лицом принимающим решение («начальник всегда прав») или этим лицом и консультантами («система коллективной ответственности»);
* определяются из статистики, если она есть в достаточном объёме и надёжна («статистика знает всё»).

К основным методам данной группы относятся следующие: метод взвешенных сумм и его аналоги, метод матрицы решений, метод дерева решений и ряд других.

Данный метод хорош тем, что лицо, принимающее решение, может установить удобные для него критерии и параметры. В конце будет получен результат с учетом предпочтений, которые были указаны на начальной стадии. Таким образом, данный метод можно использовать для разных заказчиков с разными предпочтениями.

## Анализ метода компенсации

Рациональная модель принятия решений, в которой выбор систематически оценивается по различным критериям. Более выгодные атрибуты альтернативы могут компенсировать менее выгодные. Для этого необходимо следовать систематической процедуре принятия решений компенсирующей модели, поскольку положительный балл по одному атрибуту может перевесить отрицательный результат по другому атрибуту.

Решение компенсации подразумевает, что некоторый продавец «торгует» хорошими и плохими атрибутами продукта. Например, автомобиль может иметь низкую цену и малый пробег, но медленное ускорение. Либо автомобиль имеет быстрое ускорение, но в таком случае цена будет выше, и топлива он будет потреблять больше.

Иногда решение будет включать в себя некомпенсирующую стратегию. Например, производитель может отказаться от всех безалкогольных напитков, которые содержат искусственные подсластители. Здесь другие хорошие функции, такие как вкус и низкие калории, этот атрибут «не подлежит обсуждению».

Объем усилий, которые потребитель ставит на поиск, зависит от ряда таких факторов, как рынок (количество конкурентов и насколько велики различия между брендами, которые, как ожидается, будут), характеристики продукта (важность продукта, его сложность, очевидность его качества), потребительские характеристики (интерес потребителя, правильность анализа характеристик продукта и совершения наилучшей возможной сделки) и т.д.

По идее, это наиболее простой метод, при котором человек выписывает достоинства и недостатки каждой из альтернатив и, вычеркивая попарно эквивалентные достоинства (недостатки), изучает то, что осталось и делает выбор.

Переход от сравнения качеств по различным критериям к сравнению альтернатив может быть осуществлен различными путями. Среди них следует выделить построение кривых безразличия и сравнение разностей оценок альтернатив по критериям.

Методы построения кривых или поверхностей безразличия очень трудоемки и малопригодны при количестве критериев n≥3. Для оставшихся альтернатив проще использовать метод анализа иерархий (МАИ) или один из прямых методов.

## Анализ метода анализа иерархий

Метода анализа иерархий (МАИ) был разработан для оптимизации принятия решений, когда перед человеком стоит сочетание качественных, количественных и иногда противоречивых факторов, которые учитываются в ходе анализа. МАИ очень эффективен в принятии сложных, зачастую необратимых решений.

МАИ использует решения лиц, принимающих решения, для разложения проблем в иерархии. Сложность представляется количеством уровней в иерархии, построенной с учётом модели принятия решений. Иерархия используется для получения отношений между объектами, а также относительной ценности. МАИ использует матричную алгебру для определения коэффициентов, чтобы получить математически оптимальное решение. МАИ использовался в многомиллиардных решениях.

МАИ выводит коэффициенты соотношения парных сравнений факторов и вариантов выбора. Типичные ситуации, в которых используется МАИ, относятся к:

* установке приоритетов факторов и требований, которые влияют на разработку и производительность программного обеспечения;
* выбору среди нескольких стратегий улучшения функций безопасности в автомобилях;
* выбору необходимых программных компонентов у нескольких поставщиков программного обеспечения;
* оценке качества исследовательских или инвестиционных предложений и т.д.

Метод анализа иерархий на сегодняшний день является наиболее оптимальным, распространенным и удобным. С его помощью, не затрачивая больших усилий, можно принимать решения в условиях риска и неопределенности, т.к. каждая альтернатива будет проанализирована и вероятность выбора худшей из них крайне мала. Недостатком является громоздкость построения иерархии.

# Описание разрабатываемого алгоритма

## Укрупненная схема

Укрупненная схема представлена на рисунке 1.



Рисунок 1 - Укрупненная схема алгоритма

## Развернутая схема

Развернутые схемы представлена на рисунках 2-3.



Рисунок 2 – Блок-схема функции расчета вектора приоритетов



Рисунок 3 - Блок-схема функции синтеза альтернатив

# Решение контрольного примера

На первом этапе МАИ осуществляется представление проблемы в виде иерархии. Пусть имеется 3 альтернативы. Они составляют третий низший уровень иерархии. Также имеется 5 критериев, по которым выбираются 3 альтернативы.

На втором этапе необходимо установить приоритеты критериев и произвести оценку каждой из альтернатив по критериям. Составим матрицу для попарного сравнения критериев на втором уровне по отношению к общей цели, расположенной на первом уровне (Таблицы .

Таблица 1 - Матрица попарного сравнения критериев

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Цель | К1 | К2 | К3 | К4 | К5 | q1 |
| К1 | 1 | 3 | 1/6 | 1/7 | 1/6 | 0.0603 |
| К2 | 1/3 | 1 | 1/5 | 1/4 | 1/6 | 0.045 |
| К3 | 6 | 5 | 1 | 1/5 | 3 | 0.2608 |
| К4 | 7 | 4 | 5 | 1 | 1 | 0.393 |
| К5 | 6 | 6 | 1/3 | 1 | 1 | 0.2405 |

Далее заполняем Матрица попарного сравнения альтернатив по отношению к критериям.

Таблица 2 - Матрица по отношению к К1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| К1 | А1 | А2 | А3 | q2 |
| А1 | 1 | 7 | 9 | 0.76 |
| А2 | 1/7 | 1 | 7 | 0.191 |
| А3 | 1/9 | 1/7 | 1 | 0.048 |

Таблица 3 - Матрица по отношению к К2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| К2 | А1 | А2 | А3 | q3 |
| А1 | 1 | 5 | 2 | 0.584 |
| А2 | 1/5 | 1 | 7 | 0.303 |
| А3 | 1/2 | 1/7 | 1 | 0.112 |

Таблица 4 - Матрица по отношению к К3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| К3 | А1 | А2 | А3 | q4 |
| А1 | 1 | 1/7 | 1/5 | 0.066 |
| А2 | 7 | 1 | 5 | 0.714 |
| А3 | 5 | 1/5 | 1 | 0.218 |

Таблица 5 - Матрица по отношению к К4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| К4 | А1 | А2 | А3 | q5 |
| А1 | 1 | 1 | 1 | 0.333 |
| А2 | 1 | 1 | 1 | 0.333 |
| А3 | 1 | 1 | 1 | 0.333 |

Таблица 6 - Матрица по отношению к К5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| К5 | А1 | А2 | А3 | q6 |
| А1 | 1 | 7 | 1 | 0.556 |
| А2 | 1/7 | 1 | 7 | 0.291 |
| А3 | 1 | 1/7 | 1 | 0.152 |

В завершение, осуществляется непосредственно синтез альтернатив, вычисляются приоритеты всех трех альтернатив по формуле:

Q1 = 0.0603258 \* 0.76076 + 0.0450919 \* 0.584168 + 0.260864 \* 0.0667961 + 0.393174 \* 0.333333 + 0.240544 \* 0.556782 = 0.354648

Q2 = 0.0603258 \* 0.191191 + 0.0450919 \* 0.303329 + 0.260864 \* 0.71471 + 0.393174 \* 0.333333 + 0.240544 \* 0.291062 = 0.412725

Q3 = 0.0603258 \* 0.0480493 + 0.0450919 \* 0.112502 + 0.260864 \* 0.218494 + 0.393174 \* 0.333333 + 0.240544 \* 0.152155 = 0.232627

Максимальный приоритет Q = 0.412725 и соответствует второй альтернативе.

# Листинг

#include <iostream>

#include <math.h>

#include <vector>

#include <fstream>

#include <string>

#include <algorithm>

#include <sstream>

#include <iomanip>

using namespace std;

typedef vector<vector<float>> Matrix;

struct MatrixWithVector

{

Matrix matrix;

vector<float> vectorPriority;

};

long numberOfDigits(double n) {

std::ostringstream strs;

strs << n;

return strs.str().size();

}

void printMatrix(const Matrix& M) {

long max\_len\_per\_column[M.size()];

long n = M.size(), m = M[0].size();

for (long j = 0; j < m; ++j) {

long max\_len {};

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

long num\_length {numberOfDigits(M[i][j])};

if (num\_length > max\_len)

max\_len = num\_length;

}

max\_len\_per\_column[j] = max\_len;

}

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

for (long j = 0; j < m; ++j)

std::cout << (j == 0 ? "\n| " : "") << std::setw(max\_len\_per\_column[j]) << M[i][j] << (j == m - 1 ? " |" : " ");

std::cout << '\n';

}

void initializeMatrix(Matrix& mat) {

for (int i=0; i<mat.size(); i++){

for (int j=0; j<mat.size(); j++){

mat[i][j] = 1;

}

}

}

float parseNumber(string line) {

return (float) (line[0]-'0') / (float) (line[2]-'0');

}

Matrix enterMatrix(int numberParams) {

Matrix matrix(numberParams, vector<float>(numberParams));

string number;

initializeMatrix(matrix);

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

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

cout << "Relation " << i+1 << "\\" << j+1 << endl;

cin >> number;

if (number.size() == 1){

matrix[i][j] = stof(number);

matrix[j][i] = 1/stof(number);

}

else {

float numberFromInput = parseNumber(number);

matrix[i][j] = numberFromInput;

matrix[j][i] = 1/numberFromInput;

}

}

}

return matrix;

}

void createVectorPriority(MatrixWithVector& matrix) {

vector<float> vectorGeometricMean(matrix.matrix.size());

for (int i = 0; i < matrix.matrix.size(); i++) {

float geometricMean = 1.0;

int size = matrix.matrix[0].size();

for (int j = 0; j < size; j++) {

geometricMean \*= matrix.matrix[i][j];

}

vectorGeometricMean[i] = pow(geometricMean, (1/ (float) size));

}

float normalizingFactor = 0.0;

for (int i = 0; i < vectorGeometricMean.size(); ++i) {

normalizingFactor += vectorGeometricMean[i];

}

matrix.vectorPriority = vector<float>(matrix.matrix.size());

for (int i = 0; i < matrix.matrix.size(); i++) {

matrix.vectorPriority[i] = vectorGeometricMean[i]/normalizingFactor;

}

}

vector<float> synthesisAlternatives(vector<MatrixWithVector>& matrixes) {

vector<float> result = vector<float>(matrixes[1].vectorPriority.size());

for (int i = 0; i < matrixes[1].vectorPriority.size(); ++i) {

result[i] = 0;

for (int j = 0; j < matrixes[0].vectorPriority.size(); ++j) {

result[i] += matrixes[0].vectorPriority[j] \* matrixes[j+1].vectorPriority[i];

}

cout << endl;

}

return result;

}

int main() {

cout << "Enter number criteria" << endl;

int numberK; cin >> numberK;

cout << "Enter number of alternative" << endl;

int numberA; cin >> numberA;

vector<MatrixWithVector> listAllMatrix(numberK+1);

cout << "Enter matrix of criteria" << endl;

listAllMatrix[0].matrix = enterMatrix(numberK);

printMatrix(listAllMatrix[0].matrix);

createVectorPriority(listAllMatrix[0]);

for (int j = 0; j < listAllMatrix[0].vectorPriority.size(); j++) {

cout << "q" << j << " - " << listAllMatrix[0].vectorPriority[j] << "\t";

}

cout << endl;

for (int i = 1; i < numberK+1; i++) {

cout << endl << "Enter matrix of alternative on criteria " << i << endl;

listAllMatrix[i].matrix = enterMatrix(numberA);

createVectorPriority(listAllMatrix[i]);

printMatrix(listAllMatrix[i].matrix);

for (int j = 0; j < listAllMatrix[i].vectorPriority.size(); j++) {

cout << "q" << j << " - " << listAllMatrix[i].vectorPriority[j] << "\t";

}

cout << endl;

}

cout << "Full matrixes" << endl;

for (int i = 0; i < numberK+1; ++i) {

printMatrix(listAllMatrix[i].matrix);

for (int j = 0; j < listAllMatrix[i].vectorPriority.size(); j++) {

cout << "q" << j << " - " << listAllMatrix[i].vectorPriority[j] << "\t";

}

cout << endl;

}

vector<float> resultAlternative = synthesisAlternatives(listAllMatrix);

cout << "Array of priorities of all alternatives" << endl;

for (int i = 0; i < resultAlternative.size(); i++) {

cout << "Priority of " << i+1 << " alternative is " << resultAlternative[i] << endl;

}

cout << endl << "Max priority is " << \*max\_element(resultAlternative.begin(), resultAlternative.end()) << endl;

return 0;

}

# Результат машинного решения

Enter number criteria

5

Enter number of alternative

3

Enter matrix of criteria

Relation 1\2

3

Relation 1\3

1/6

Relation 1\4

1/7

Relation 1\5

1/6

Relation 2\3

1/5

Relation 2\4

1/4

Relation 2\5

1/6

Relation 3\4

1/5

Relation 3\5

3

Relation 4\5

1

| 1 3 0.166667 0.142857 0.166667 |

| 0.333333 1 0.2 0.25 0.166667 |

| 6 5 1 0.2 3 |

| 7 4 5 1 1 |

| 6 6 0.333333 1 1 |

q0 - 0.0603258 q1 - 0.0450919 q2 - 0.260864 q3 - 0.393174 q4 - 0.240544

Enter matrix of alternative on criteria 1

Relation 1\2

7

Relation 1\3

9

Relation 2\3

7

| 1 7 9 |

| 0.142857 1 7 |

| 0.111111 0.142857 1 |

q0 - 0.76076 q1 - 0.191191 q2 - 0.0480493

Enter matrix of alternative on criteria 2

Relation 1\2

5

Relation 1\3

2

Relation 2\3

7

| 1 5 2 |

| 0.2 1 7 |

| 0.5 0.142857 1 |

q0 - 0.584168 q1 - 0.303329 q2 - 0.112502

Enter matrix of alternative on criteria 3

Relation 1\2

1/7

Relation 1\3

1/5

Relation 2\3

5

| 1 0.142857 0.2 |

| 7 1 5 |

| 5 0.2 1 |

q0 - 0.0667961 q1 - 0.71471 q2 - 0.218494

Enter matrix of alternative on criteria 4

Relation 1\2

1

Relation 1\3

1

Relation 2\3

1

| 1 1 1 |

| 1 1 1 |

| 1 1 1 |

q0 - 0.333333 q1 - 0.333333 q2 - 0.333333

Enter matrix of alternative on criteria 5

Relation 1\2

7

Relation 1\3

1

Relation 2\3

7

| 1 7 1 |

| 0.142857 1 7 |

| 1 0.142857 1 |

q0 - 0.556782 q1 - 0.291062 q2 - 0.152155

Full matrixes

| 1 3 0.166667 0.142857 0.166667 |

| 0.333333 1 0.2 0.25 0.166667 |

| 6 5 1 0.2 3 |

| 7 4 5 1 1 |

| 6 6 0.333333 1 1 |

q0 - 0.0603258 q1 - 0.0450919 q2 - 0.260864 q3 - 0.393174 q4 - 0.240544

| 1 7 9 |

| 0.142857 1 7 |

| 0.111111 0.142857 1 |

q0 - 0.76076 q1 - 0.191191 q2 - 0.0480493

| 1 5 2 |

| 0.2 1 7 |

| 0.5 0.142857 1 |

q0 - 0.584168 q1 - 0.303329 q2 - 0.112502

| 1 0.142857 0.2 |

| 7 1 5 |

| 5 0.2 1 |

q0 - 0.0667961 q1 - 0.71471 q2 - 0.218494

| 1 1 1 |

| 1 1 1 |

| 1 1 1 |

q0 - 0.333333 q1 - 0.333333 q2 - 0.333333

| 1 7 1 |

| 0.142857 1 7 |

| 1 0.142857 1 |

q0 - 0.556782 q1 - 0.291062 q2 - 0.152155

Array of priorities of all alternatives

Priority of 1 alternative is 0.354648

Priority of 2 alternative is 0.412725

Priority of 3 alternative is 0.232627

Max priority is 0.412725

# Заключение

Принятие важных решений является неотъемлемой частью не только крупных компаний, но и обычных людей.

В данной работе был выделен объект исследования, проанализирована предметная область, определены цели и задачи работы. Проведен анализ существующих методов многокритериального анализа и реализован наиболее подходящий для реализации цели выпускной квалификационной работы.

При анализе МКА было выявлено, что для решения задач работы потребуется написать свою программную реализацию алгоритма метода анализа иерархий, а не использовать существующие, так как существующие разработки были сделаны для других конкретных целей, поэтому и наша разработка будет носить такой же характер.

Была спроектирована программа и реализована на языке C++.

# Список литературы

1. Борисов А.Н. Обработка нечеткой информации в системах принятия решений / Алексеев А.В., Меркурьева Г.В., Сладзь Н.Н., Глушков В.И. – М.: Радио и связь, 2003.
2. Семенов С.С. Обзор методов принятия решений при разработке сложных технических систем // Функциональная надежность. Теория и практика. / Полтавский А.В., Маклаков В.В., Крянев А.В. – Надежность, 2014.