Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

**ЛАБОРАТОРНАЯ РАБОТА № 5**

**ИССЛЕДОВАНИЕ МЕТОДОВ РЕШЕНИЯ МНОГОКРИТЕРИАЛЬНЫХ ЗАДАЧ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ ПОСТРОЕНИЯ МНОЖЕСТВА ПАРЕТО**

|  |  |
| --- | --- |
|  |  |

Выполнил:

Ст. гр. ИТб-33

Лисянский А. И.

Проверил:

Кротов К. В.

Севастополь

2014

**Цель работы**

Исследовать способы формирования множества Парето-оптимальных решений и определения эффективных решений в этом множестве.

**Программа выполнения работы**

Для первого и третьего вариантов в соответствии с заданием необходимо реализовать следующий порядок действий для выполнения лабораторной работы:

а) разработать процедуру определения на основе задаваемого множества решений *Х* и соответствующих им значений критериев  и  множества *Р(Х)*, представляющего собой Парето-границу *Х*;

б) разработать процедуру определения координат идеальной точки  (точки утопии);

в) разработать процедуру расчета расстояния  до точки утопии для координат текущего рассматриваемого решения ;

г) разработать процедуру определения эффективного решения , расстояние до которого от идеальной точки  является минимальным.

**Вариант задания**

Вариант 1. Требуется для задаваемого множества *Х* в виде:  выполнить определение эффективных решений двухкритериальной задачи выбора с использованием метода идеальной точки. Значения критериев  и  для соответствующих решений  () сведены в матрицу, представленную ниже.



**Аналитическое выполнение**

x2>x1

x2>x8

x2>x10

x4>x3>x1

x6>x1

x6>x8

x6>x10

x7>x3>x1

x7>x5

x9>x3>x1

x9>x5

Соответственно x2,x4,x6,x7 являются Парето-границей

Если расчитать расстояние между решениями и идеальной точкой, то минимальное значение будет у x7.

Рисунок 1 – Решения на критериальном пространстве

**Код программы**

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

#include <math.h>

using namespace std;

const int N=10;

struct reshenie{

int f1,f2;

bool inPareto;

};

bool b\_r(reshenie a,reshenie b)

{

if(a.f1 > b.f1 && a.f2 >= b.f2 ||

a.f1 >= b.f1 && a.f2 > b.f2)

return true;

else

return false;

}

int main(){

reshenie matrix[N];

int i;

ifstream k("5.txt", ios::in);

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

matrix[i].inPareto = false;

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

k >> matrix[i].f1 >> matrix[i].f2;

int t;

for (t=0; t<N-1; t++)

for(i=0; i<t-1; i++)

{

if(b\_r(matrix[t],matrix[i]) == true && b\_r(matrix[i],matrix[t]) == false)

{

matrix[t].inPareto = true;

matrix[i].inPareto = false;

}

if(b\_r(matrix[t],matrix[i]) == false && b\_r(matrix[i],matrix[t]) == true)

{

matrix[i].inPareto = true;

matrix[t].inPareto = false;

}

if(b\_r(matrix[t],matrix[i]) == false && b\_r(matrix[i],matrix[t]) == false)

{

matrix[t].inPareto = true;

matrix[i].inPareto = true;

}

}

cout << "Решения, вошедшие в Паретто-границу" <<endl;

for(i=1; i<N; i++)

{

if(matrix[i].inPareto == true)

{

cout << "x" << i+1 <<" ";

}

}

cout << endl;

int maxf1 = matrix[0].f1,maxf2 = matrix[0].f2;

for(i=1; i<N; i++)

{

if(matrix[i].f1 > maxf1)

maxf1=matrix[i].f1;

if(matrix[i].f2 > maxf2)

maxf2=matrix[i].f2;

}

double minf1f2=999;

int index;

for(i=1; i<N; i++)

{

if(matrix[i].inPareto == true)

{

double temp;

temp=sqrt((maxf1 -matrix[i].f1)\*(maxf1 -matrix[i].f1)+(maxf2 -matrix[i].f2)\*(maxf2 -matrix[i].f2));

if(temp<minf1f2)

{

minf1f2=temp;

index=i;

}

}

}

cout << "Наиболее предпочтительное решение на Паретто-границе x" << index+1 << endl;

return 0;

}

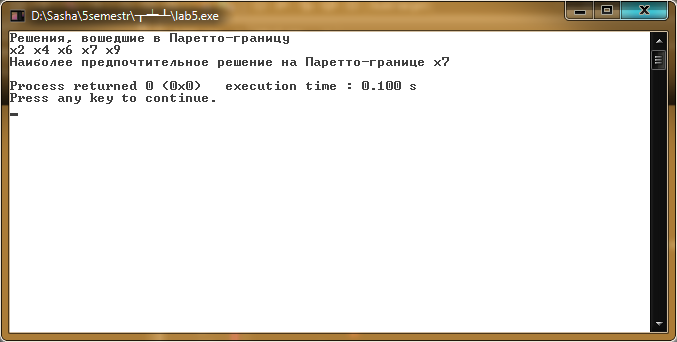


Рисунок 2 – результат выполнения программы

**Вывод**

В ходе выполнения лабораторной работы была написана программа определения эффективного решения . Были изучены основные правила построения Парето-границы, определения эффективных решений, входящих в Парето-границу, определений эффективного решения .