МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

кафедра Информационные системы

Сирота Марина Романовна

Институт информационных технологий и управления в технических системах

курс 3 группа ИС/б-32-о

09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ

по лабораторной работе №5

по дисциплине: «Теория принятия решений»

по теме: «Исследование методов решения многокритериальных задач принятия решений на основе построения множества Парето»

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

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

доц. Кротов К.В.

(должность) (подпись) (инициалы, фамилия)

Севастополь

2018

1. ЦЕЛЬ РАБОТЫ

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

1. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

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.

1. ВЫПОЛНЕНИЕ РАБОТЫ
   1. Код программы.

#include <iostream>

#include <fstream>

#include <cmath>

#include <iomanip>

using namespace std;

int \*\*read(int &n, int &m) {

ifstream is("input.txt");

is >> n >> m;

int \*\*matrix = new int \*[10];

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

matrix[i] = new int[m + 1];

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

is >> matrix[i][j];

}

matrix[i][m] = 1;

}

return matrix;

}

void pareto(int n, int m, int \*\*matrix) {

int gt, lt, et;

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

if (matrix[i][m] == 0) {

continue;

}

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

if (matrix[j][m] == 0) {

continue;

}

et = gt = lt = 0;

for (int k = 0; k < m; k++) {

if (matrix[i][k] == matrix[j][k]) {

et++;

}

else if (matrix[i][k] > matrix[j][k]) {

gt++;

}

else {

lt++;

}

}

if (gt > 0 && lt == 0) {

matrix[j][m] = 0;

}

if (gt == 0 && lt > 0) {

matrix[i][m] = 0;

}

}

}

}

int \*utopia(int n, int m, int \*\*matrix) {

int \*result = new int[m];

int max;

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

max = matrix[0][i];

for (int j = 1; j < n; j++) {

if (matrix[j][i] > max) {

max = matrix[j][i];

}

}

result[i] = max;

}

return result;

}

double \*distance(int n, int m, int \*\*matrix, int \*utopia\_point) {

double \*rx = new double[n];

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

rx[i] = 0;

if (matrix[i][m] == 0) {

continue;

}

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

rx[i] += pow((utopia\_point[j] - matrix[i][j]), 2);

}

rx[i] = sqrt(rx[i]);

}

return rx;

}

int effective(int n, int m, double \*rx) {

int result = 0;

double min;

min = 0;

bool was\_set = false;

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

if (rx[i] == 0) {

continue;

}

if (!was\_set || min > rx[i]) {

was\_set = true;

min = rx[i];

result = i + 1;

}

}

return result;

}

char \*concat(char c, int d) {

char \*result = new char[4];

sprintf(result, "%c%d", c, d);

return result;

}

void print(int n, int m, int \*\*matrix) {

cout << setw(4) << "";

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

cout << setw(3) << concat('f', i + 1) << "|";

}

cout << endl;

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

cout << setw(3) << concat('x', i + 1) << ":";

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

cout << setw(3) << matrix[i][j] << "|";

}

cout << endl;

}

}

int main() {

setlocale(LC\_ALL, "RUS");

int n, m;

int \*\*matrix = read(n, m);

cout << " Значения критериев:" << endl;

print(n, m, matrix);

cout << endl;

int cout\_width = (int)ceil(log10(n)) + 1;

pareto(n, m, matrix);

cout << " Решения на Парето-границе:" << endl;

bool was\_printed = false;

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

if (matrix[i][m]) {

if (was\_printed) {

cout << ", ";

}

was\_printed = true;

cout << setw(cout\_width + 1) << concat('x', i + 1);

}

}

if (!was\_printed) {

cout << "Отсутствуют" << endl;

return 0;

}

cout << endl << endl;

int \*utopia\_point = utopia(n, m, matrix);

cout << " Точка утопии" << endl;

cout << "{";

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

cout << setw(cout\_width) << utopia\_point[i];

if (i != m - 1) {

cout << ", ";

}

}

cout << "}" << endl;

double \*rx = distance(n, m, matrix, utopia\_point);

cout << " Расстояния до точки утопии" << endl;

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

if (matrix[i][m]) {

cout << "r[" << setw(cout\_width + 1) << concat('x', i + 1) << "] = "

<< setw(20) << setprecision(18) << rx[i] << endl;

}

}

cout << endl;

int effective\_solution = effective(n, m, rx);

cout << " Эффективное решение" << endl;

cout << concat('x', effective\_solution) << endl;

system("pause");

return 0;

}

* 1. Результаты выполнения программы.

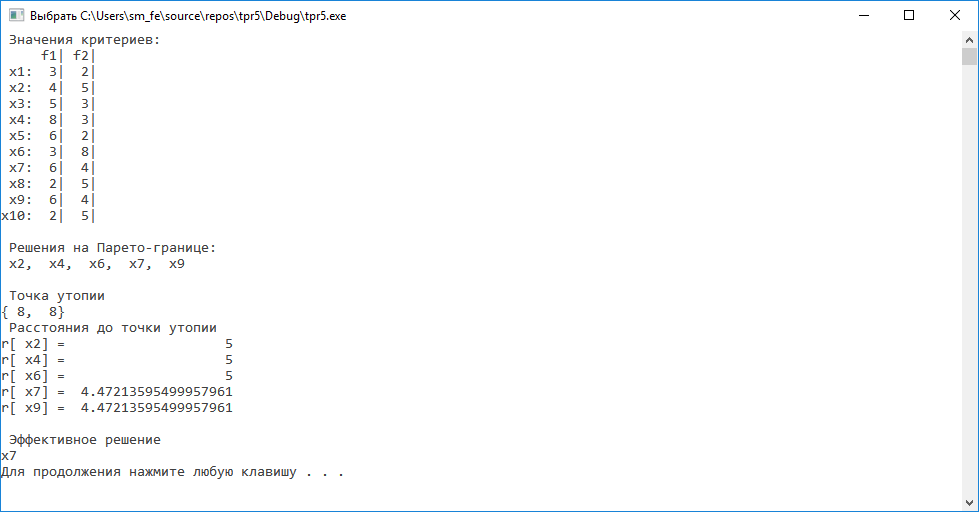


Рисунок 3.1 – Результаты выполнения программы

ВЫВОДЫ

В ходе пятой лабораторной работы по дисциплине «Теория принятия решений» были исследованы способы формирования множества Парето-оптимальных решений и определения эффективных решений в этом множестве.