|  |
| --- |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **"МИРЭА – Российский технологический университет"**  **РТУ МИРЭА** |

Институт кибербезопасности и цифровых технологий

Кафедра КБ-2 «Прикладные информационные технологии»

**ОТЧЕТ   
о выполнении контрольного домашнего задания №1.1**

по дисциплине**:  
«Формализованные модели и методы решения аналитических задач»**

Выполнила: студентка 3 курса

Семёнова С.А. БИСО-03-20

Вариант 13

Проверил:*\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Москва 2023 г



**Решение**

По варианту принимаем переменную n равную 13. Генерируем случайным образом 20 точек в пространстве критериев, заданном системой ограничений-неравенств (1). Графическое представление системы ограничений-неравенств и созданные от руки точки представлены на рисунке 1.

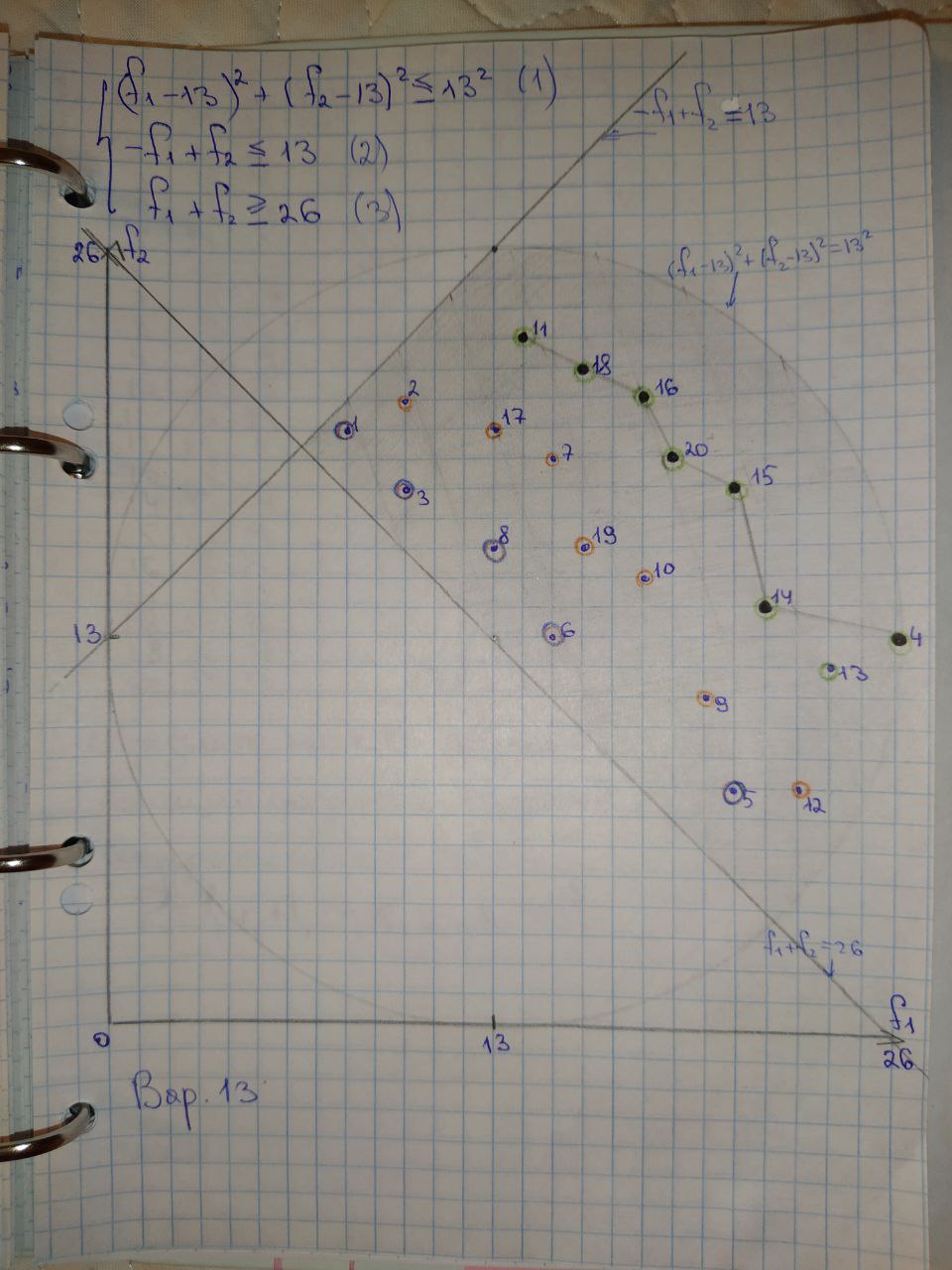


Рисунок 1 – Система ограничений-неравенств и точки

Далее необходимо определить множество эффективных проектов с помощью алгоритма исключения заведомо неэффективных решений. Для этого для каждой точки используем конус доминирования и исключаем заведомо неэффективные. Таблица выполненного алгоритма представлена на рисунке 2.

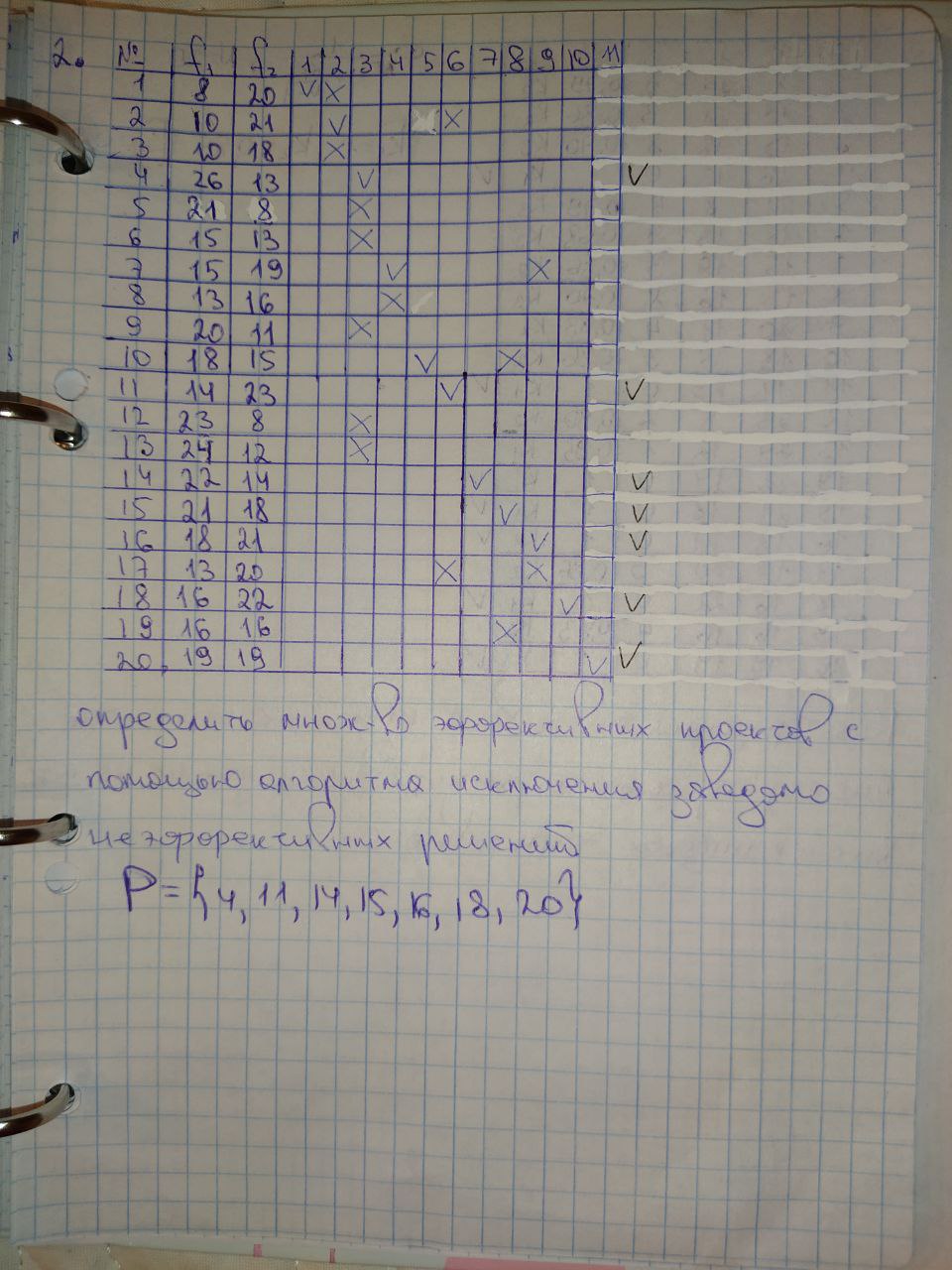


Рисунок 2 – Таблица алгоритма для задания 2

Для данной выборке 20 точек Парето-оптимальными будут точки 4, 11, 14, 15, 16, 18, 20.

Выполнение кластеризации множества проектов с помощью алгоритма многокритериального ранжирования данных на основе вычисления индекса эффективности. Таблица выполненной кластеризации представлена на рисунке 3.

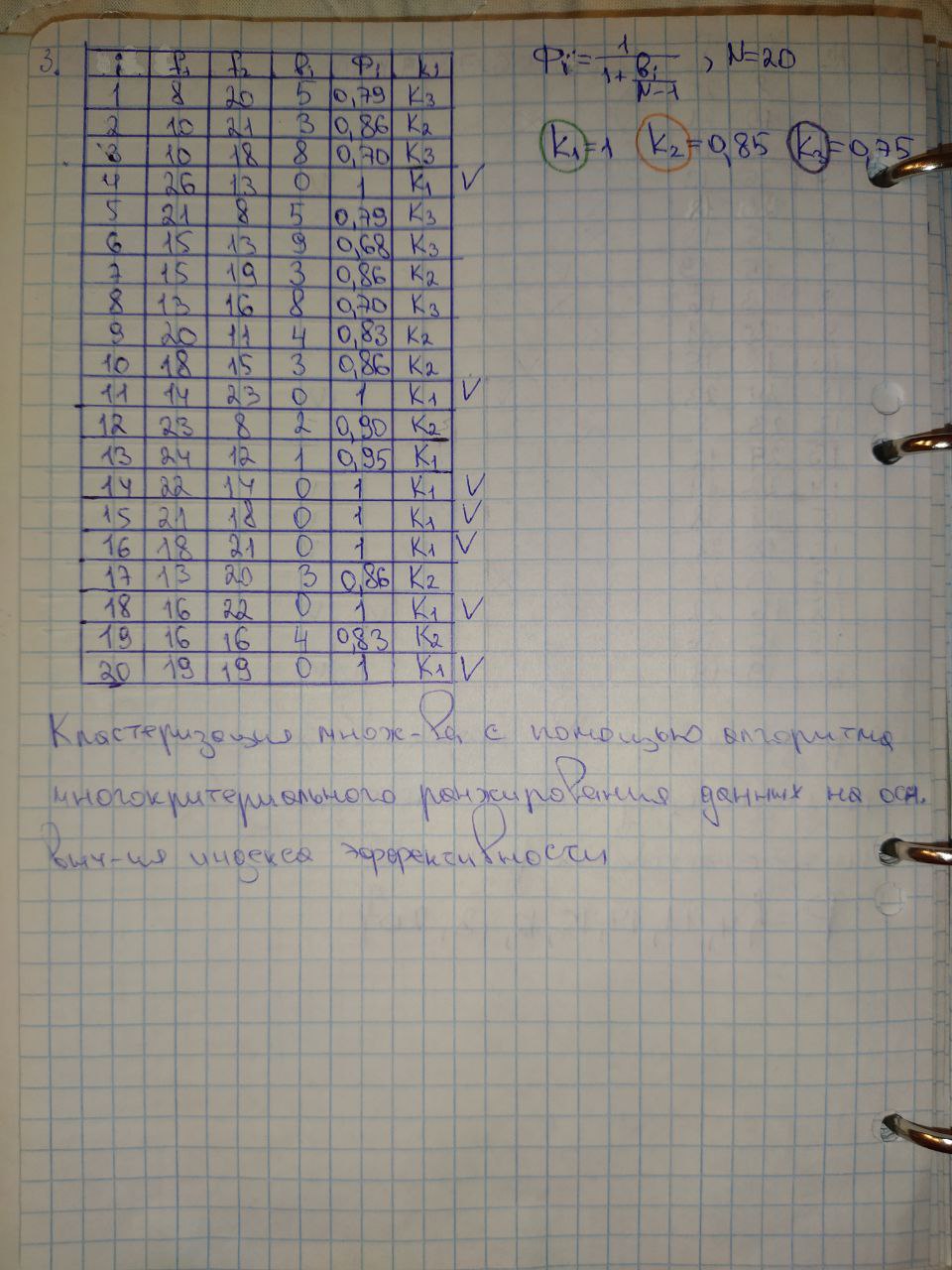


Рисунок 3 – Кластеризация

**Листинг программы**

import random

import matplotlib

from prettytable import PrettyTable

import numpy as np

import matplotlib.pyplot as mpl

matplotlib.use('TkAgg')

n = 13

num = 20

a = int(input('1 - Задать точки случайно, 2 - Показать решение по тетради\n'))

if a == 1:

num = int(input("Введите количество точек\n"))

k1 = 1

k2 = 0.85

k3 = 0.75

count = 0

lst = []

class Point:

def \_\_init\_\_(self, x, y, status='V', wasactive="-", dele='', bi=0, bname='', findex=0, fstatus=0):

self.x = x

self.y = y

self.status = status

self.wasactive = wasactive

self.dele = dele

self.bi = bi

self.bname = bname

self.findex = findex

self.fstatus = fstatus

while count < num:

if a == 1:

y = random.uniform(3.81, 26)

x = random.uniform(6.5, 26)

if (y <= n + x) and (y >= 2 \* n - x) and ((x - n) \*\* 2 + (y - n) \*\* 2 <= n \*\* 2):

lst.append(Point(x, y))

count = count + 1

else:

lst.append(Point(8, 20))

lst.append(Point(10, 21))

lst.append(Point(10, 18))

lst.append(Point(26, 13))

lst.append(Point(21, 8))

lst.append(Point(15, 13))

lst.append(Point(15, 19))

lst.append(Point(13, 16))

lst.append(Point(20, 11))

lst.append(Point(18, 15))

lst.append(Point(14, 23))

lst.append(Point(23, 8))

lst.append(Point(24, 12))

lst.append(Point(22, 14))

lst.append(Point(21, 18))

lst.append(Point(18, 21))

lst.append(Point(13, 20))

lst.append(Point(16, 22))

lst.append(Point(16, 16))

lst.append(Point(19, 19))

count = num

test\_number = 0

while True:

test\_text = input("Выберите, что вывести:\n0 - График с точками\n1 - Множество Парето на графике\n2 - Кластеры с раскраской по эффективности\n")

test\_number = int(test\_text)

if test\_number == 0:

y1 = lambda x: n + x

y2 = lambda x: (n \* 2) - x

fig, axes = mpl.subplots()

x = np.linspace(0, 40, 2)

drawing\_uncolored\_circle = mpl.Circle((n, n), n, color='c', fill=False,

label='(f1 - 13)^2 + (f2 - 13)^2 <= 169')

axes.set\_aspect(1)

axes.add\_artist(drawing\_uncolored\_circle)

mpl.plot(x, y1(x), color='b', label='-f1 + f2 <= 13')

mpl.plot(x, y2(x), color='r', label='f1 + f2 >= 26')

for i in range(num):

mpl.plot(lst[i].x, lst[i].y, marker=".", color="k")

mpl.xlabel("F1")

mpl.ylabel("F2")

mpl.legend(fontsize='xx-small', frameon=False)

mpl.suptitle('График')

mpl.savefig('Figure.png')

mpl.show()

elif test\_number == 1:

for i in range(num):

for j in range(num):

if (lst[i].status == 'V') and (i != j):

lst[i].wasactive = '+'

if (lst[j].x <= lst[i].x) and (lst[j].y <= lst[i].y) and (lst[j].status != 'X'):

lst[j].status = 'X'

j1 = j + 1

lst[i].dele = lst[i].dele + str(j1) + ' '

pt1 = PrettyTable()

pt1.field\_names = ["id", "f1", "f2", 'V/X', 'Use', 'Del']

for i in range(num):

pt1.add\_row([i + 1, lst[i].x, lst[i].y, lst[i].status, lst[i].wasactive, lst[i].dele])

print(pt1)

y1 = lambda x: n + x

y2 = lambda x: (n \* 2) - x

fig, axes = mpl.subplots()

x = np.linspace(0, 40, 2)

drawing\_uncolored\_circle = mpl.Circle((n, n), n, color='c', fill=False,

label='(f1 - 13)^2 + (f2 - 13)^2 <= 169')

axes.set\_aspect(1)

axes.add\_artist(drawing\_uncolored\_circle)

mpl.plot(x, y1(x), color='b', label='-f1 + f2 <= 13')

mpl.plot(x, y2(x), color='r', label='f1 + f2 >= 26')

for i in range(num):

if lst[i].status == 'V':

mpl.plot(lst[i].x, lst[i].y, marker=".", color="k")

mpl.xlabel("F1")

mpl.ylabel("F2")

mpl.suptitle('Множество Парето')

mpl.savefig('Figure2.png')

mpl.show()

elif test\_number == 2:

y1 = lambda x: n + x

y2 = lambda x: (n \* 2) - x

fig, axes = mpl.subplots()

x = np.linspace(0, 40, 2)

drawing\_uncolored\_circle = mpl.Circle((n, n), n, color='c', fill=False, label='(f1 - 20)^2 + (f2-20)^2 <= 169')

axes.set\_aspect(1)

axes.add\_artist(drawing\_uncolored\_circle)

mpl.plot(x, y1(x), color='b', label='-f1 + f2 <= 13')

mpl.plot(x, y2(x), color='r', label='f1 + f2 >= 26')

for i in range(num):

for j in range(num):

if i != j:

if (lst[j].x >= lst[i].x) and (lst[j].y >= lst[i].y):

lst[i].bi = lst[i].bi + 1

j1 = j + 1

lst[i].bname = lst[i].bname + str(j1) + ' '

lst[i].findex = 1 / (1 + (lst[i].bi / (num - 1)))

if (abs(k1 - lst[i].findex) < abs(k2 - lst[i].findex)) and (

abs(k1 - lst[i].findex) < abs(k3 - lst[i].findex)):

lst[i].fstatus = 1

elif (abs(k2 - lst[i].findex) < abs(k1 - lst[i].findex)) and (

abs(k2 - lst[i].findex) < abs(k3 - lst[i].findex)):

lst[i].fstatus = 2

elif (abs(k3 - lst[i].findex) < abs(k1 - lst[i].findex)) and (

abs(k3 - lst[i].findex) < abs(k2 - lst[i].findex)):

lst[i].fstatus = 3

pt2 = PrettyTable()

pt2.field\_names = ["id", "f1", "f2", "bi", 'bi()', 'Ф', 'Ki']

for i in range(num):

pt2.add\_row([i + 1, lst[i].x, lst[i].y, lst[i].bi, lst[i].bname, lst[i].findex, lst[i].fstatus])

print(pt2)

# Вывод кластеров по индексам эффективности

count1 = 0

count2 = 0

count3 = 0

for i in range(num):

if lst[i].fstatus == 1 and (count1 == 0):

mpl.plot(lst[i].x, lst[i].y, marker=".", color="g")

count1 = count1 + 1

for i in range(num):

if lst[i].fstatus == 2 and (count2 == 0):

mpl.plot(lst[i].x, lst[i].y, marker=".", color="y")

count2 = count2 + 1

for i in range(num):

if lst[i].fstatus == 3 and (count3 == 0):

mpl.plot(lst[i].x, lst[i].y, marker=".", color="m")

count3 = count3 + 1

for i in range(num):

if lst[i].fstatus == 1:

mpl.plot(lst[i].x, lst[i].y, marker=".", color="g")

if lst[i].fstatus == 2:

mpl.plot(lst[i].x, lst[i].y, marker=".", color="y")

if lst[i].fstatus == 3:

mpl.plot(lst[i].x, lst[i].y, marker=".", color="m")

mpl.suptitle('Кластеры')

mpl.xlabel("F1")

mpl.ylabel("F2")

mpl.legend(['1', '2', '3'], fontsize='xx-small')

mpl.savefig('Figure3.png')

mpl.show()