|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт комплексной безопасности и специального приборостроения  Кафедра КБ-2 «Прикладные информационные технологии» |

**ЗАЧТЕНО**

\_\_\_\_\_\_\_\_\_\_ /В. А. Серов/

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

**Контрольное домашнее задание № 1.5**

по дисциплине

**«Формализованные модели и методы решения аналитических задач**»

|  |  |
| --- | --- |
| **Выполнил** |  |
| Студент 3 курса:  Группы:  Специальности: | Кабанова С.О.  БИСО-01-20  10.05.04 |

Москва

2023 г.

Оглавление

[Задание 3](#_Toc135074676)

[Вариант 14 4](#_Toc135074677)

[Уточнение задания: 4](#_Toc135074678)

[Решение 5](#_Toc135074679)

[Метод идеальной точки 5](#_Toc135074680)

[Используем метод Франка-Вулфа: 6](#_Toc135074681)

[Программная реализация была выполнена с использование Jupyter Notebook. 12](#_Toc135074682)

[Выводы: 14](#_Toc135074683)

[Дополнение: 14](#_Toc135074684)

[ПРИЛОЖЕНИЕ 15](#_Toc135074685)

# Задание

Дана многокритериальная аналитическая задача:

,

,



при ограничениях:

, где  – номер варианта задания.

Решить задачу методом «идеальной точки». Использовать алгоритм Франка-Вульфа.

Начальные условия заданы в таблице.

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

# Вариант 14

## Уточнение задания:

Ограничения:

Начальные условия:

На рисунке 1 представлена визуализация облисти ограничений.

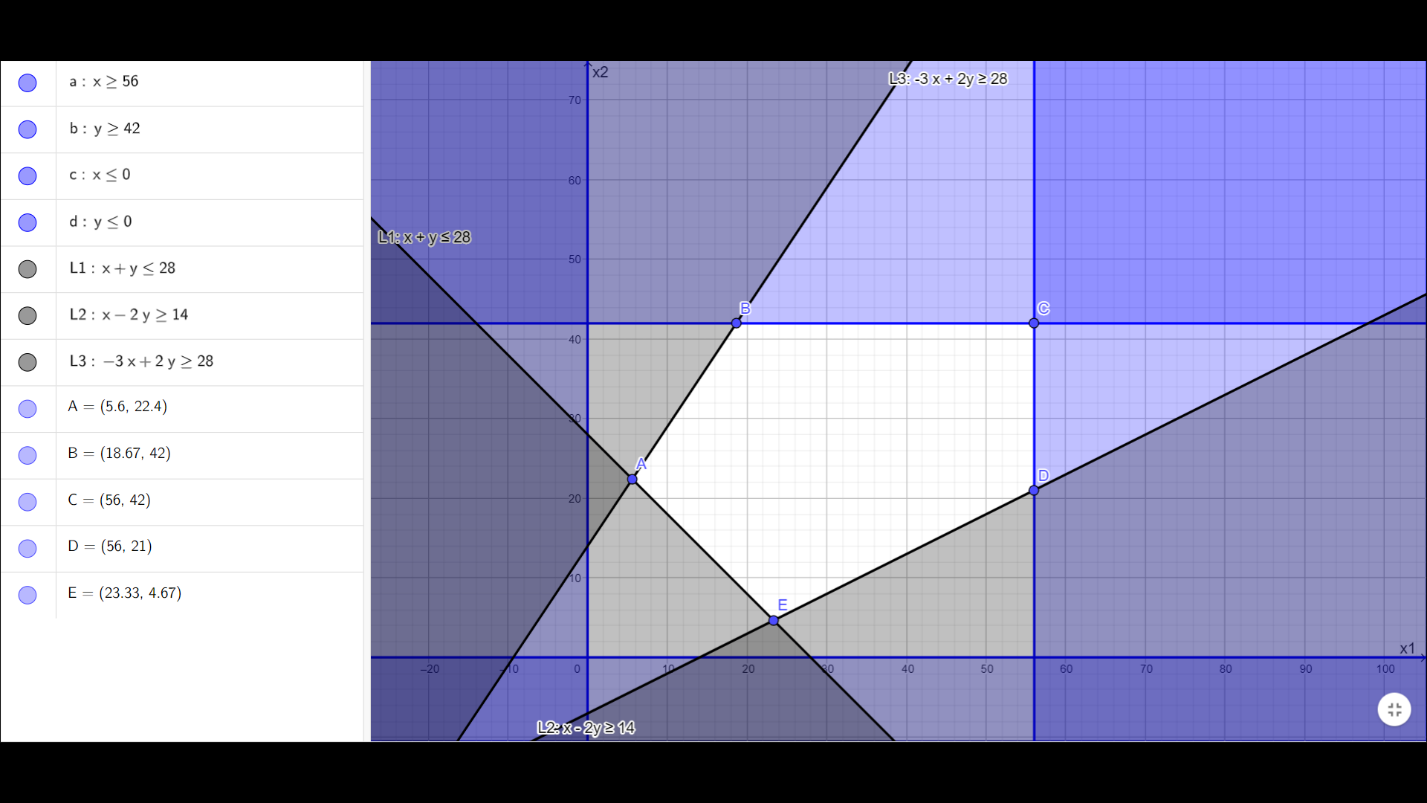


Рисунок 1 – Визуализация области ограничений D

## Решение

### Метод идеальной точки

1. Найдем оптимальные решения на области D для всех критериев:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| A | 5.6 | 22.4 | 28 | 5.6 | -61.6 |
| B | 18.67 | 42 | 60.67 | -14.01 | -107.33 |
| C | 56 | 42 | 98 | -126 | -70 |
| D | 56 | 21 | 77 | -147 | -7 |
| E | 23.33 | 4.67 | 28 | -65.32 | 9.33 |

Визуализация оптимальных точек для каждого критерия представлена на рисунке 2.

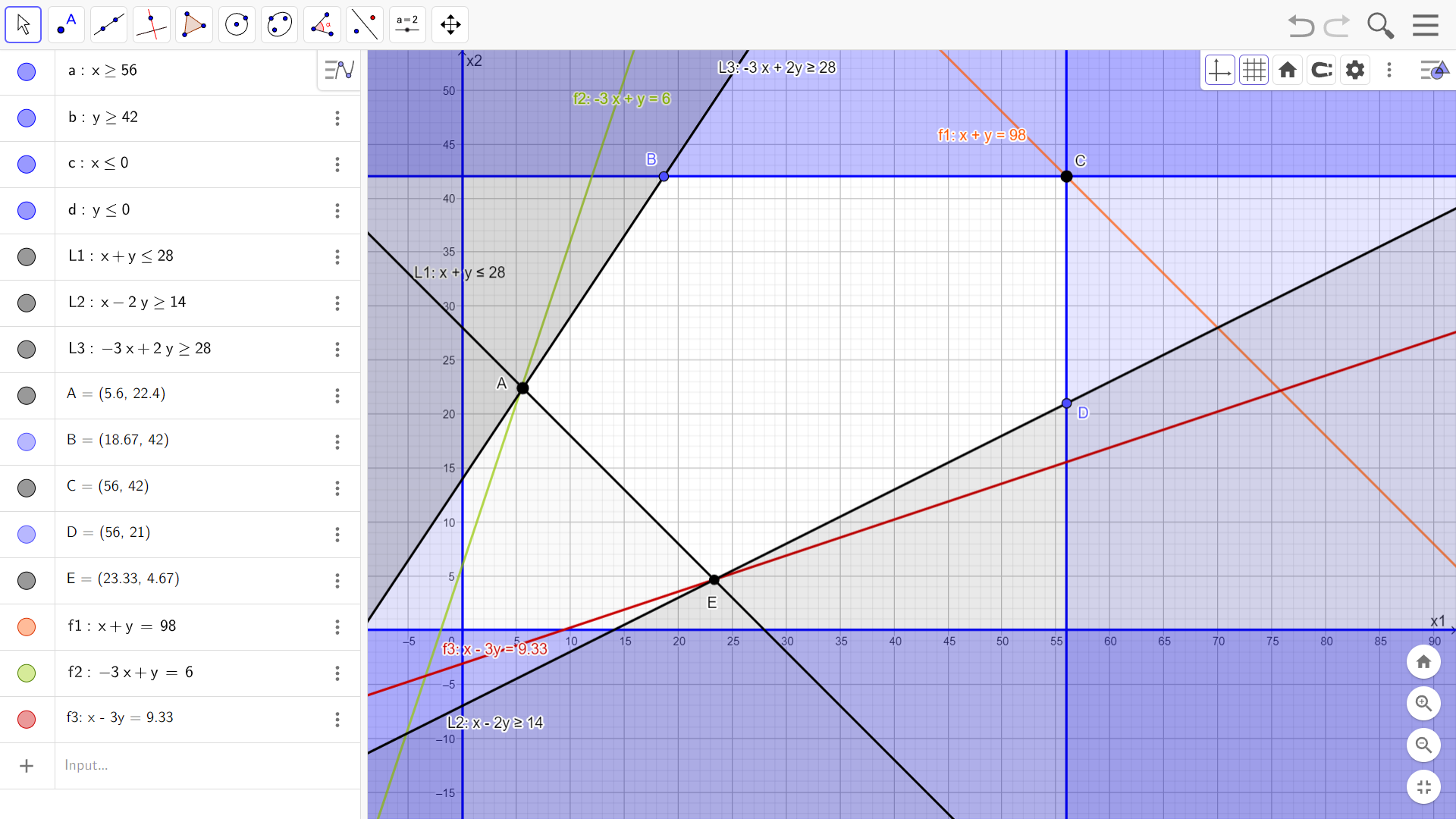


Рисунок 2 – Оптимальные решения для каждого критерия

1. Найдем расстояние от F(x) до :

**=**

*+=*

1. Определим

### Используем метод Франка-Вулфа:

#### Начальный этап

1. Задать начальное приближение

Визуализация начального приближения представлена на рисунке 3.

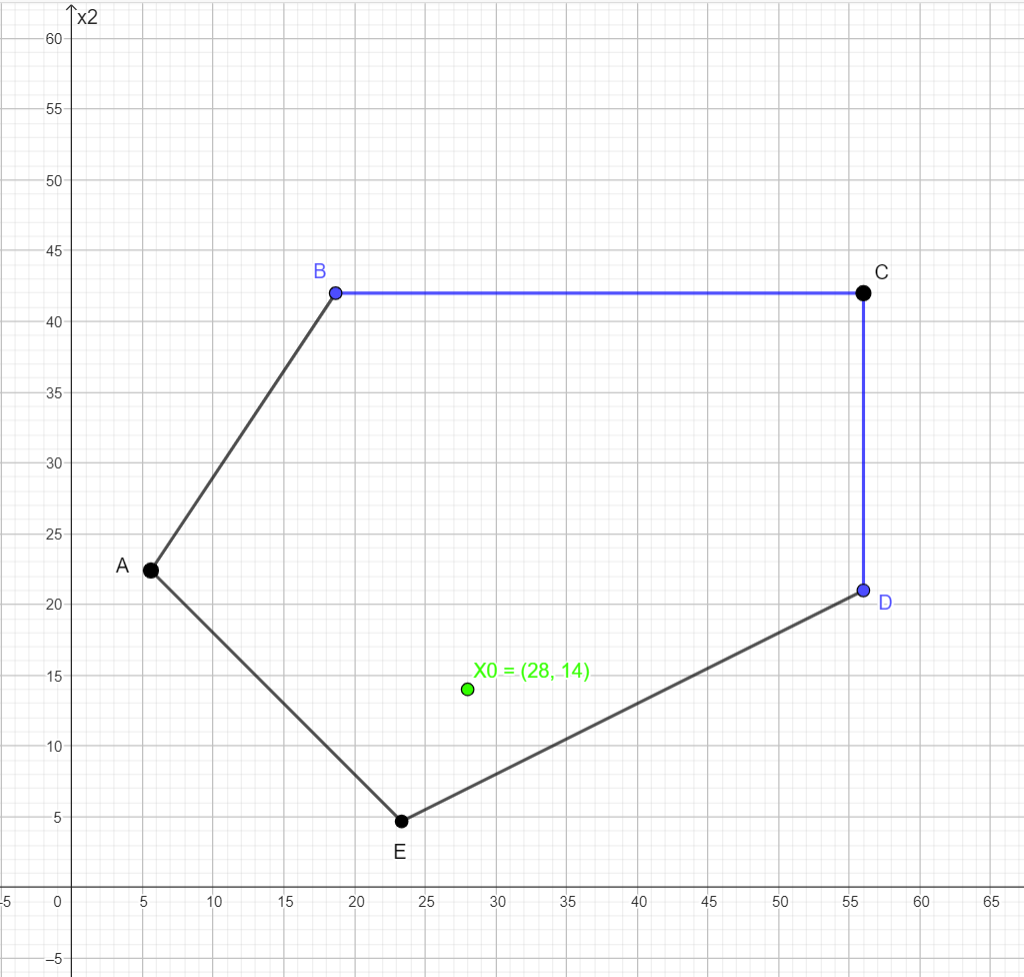


Рисунок 3 – начальное приближение на области D

Значения целевых функции в начальной точке:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | 28 | 14 | 42 | -70 | -14 |

1. Вычислить вектор градиента

Решим систему:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 1 | 2 | 3 |
|  | 14,31 | 13,38 | 27,69 | -29,55 | -25,83 |

#### Шаг 1. Составить вспомогательную функцию:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| A | 5.6 | 22.4 | -1108,46 |
| B | 18.67 | 42 | 331,2898 |
| C | 56 | 42 | 11341,4 |
| D | 56 | 21 | 13929,02 |
| E | 23.33 | 4.67 | 6305,513 |

*A*

#### Шаг 2. Приближение к решению (1)

*,*

где :

Подставим значения и в :

= 11\*(+ 11\*(

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 1 | 2 | 3 |
|  | 17,472 | 17,948 | 28,84 | -41,048 | -16,632 |

#### Шаг 3. Составить вспомогательную функцию

Определим :

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| A | 5.6 | 22.4 | -1410,19 |
| B | 18.67 | 42 | -2449,3 |
| C | 56 | 42 | -1559,21 |
| D | 56 | 21 | -111,972 |
| E | 23.33 | 4.67 | 234,4428 |

#### Шаг 4. Приближение к решению (2)

*,*

где :

Подставим значения и в :

=

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 1 | 2 | 3 |
|  | 18,20425 | 16,28825 | 34,4925 | -38,3245 | -30,6605 |

Визуализация полученных точек представлена на рисунке 4.

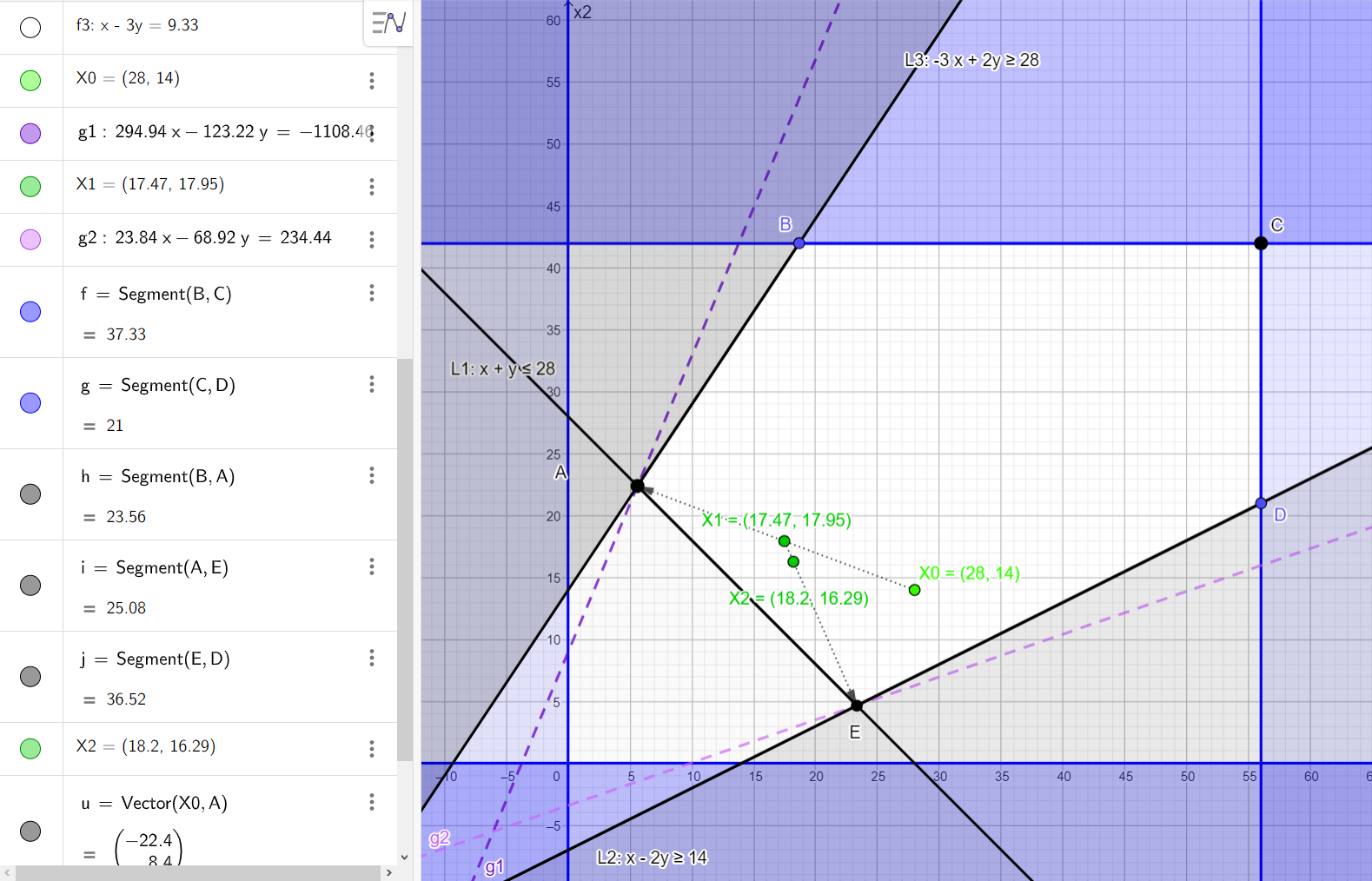


Рисунок 4 – решения на области D

Алгоритмм продолжает свою работу до тех, пока точность решения не будет удовлетворять неравенств:

Ответом для многокритериальной аналитической задачи будет точка

,

## Программная реализация была выполнена с использование Jupyter Notebook.

Визуализация исходной области D представлена на рисунке 5.

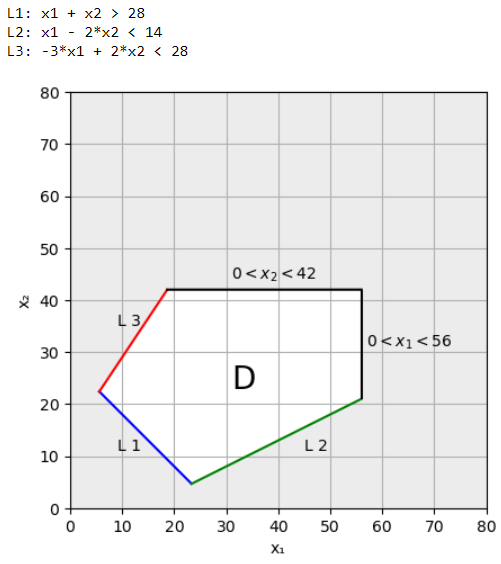


Рисунок 5 – программная визуализация области D

Выод программного решения многокритериальной аналитической задачи представлено на рисунке 6.

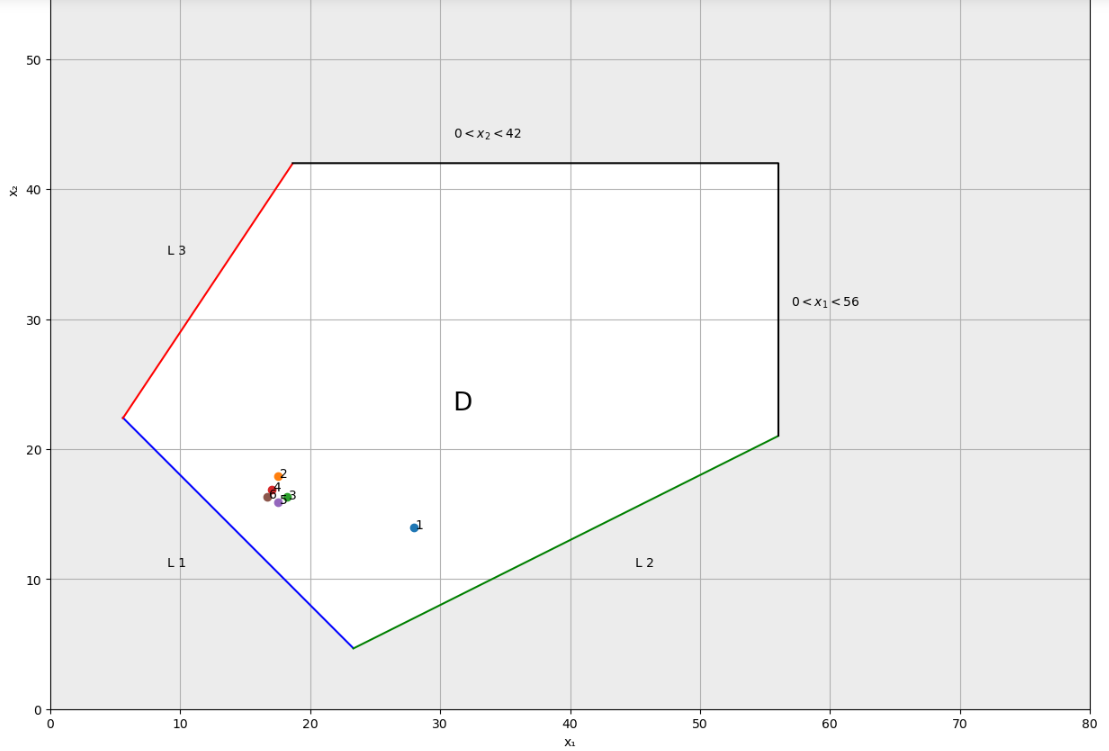


Рисунок 6 – программная визуализация решения многокритериальной аналитической задачи

# Выводы:

В ходе работы были изучены метод идеальной точки и метод Франка-Вульфа. Было получено решение многокритериалььной аналитической задачи с точность до .

По сравнению с предыдущими алгоритмами, метод Франка-Вульфа более трудоемкий, но дает более точный результат. Отличительная черта этого алгоритма – бастрая и гарантированная сходимость.

## Дополнение:

Программная реализация представлена в приложении.

# ПРИЛОЖЕНИЕ

def frank\_wolfe(funcs,d,d\_rhs,X):

x1,x2,l = symbols('x1 x2 l')

ideal\_points = []

#Создание идеальных точек

for f in funcs:

res = linprog(c=f,A\_ub=d,b\_ub=d\_rhs)

ideal\_points.append(round(res.fun\*-1,2))

print("F\*",ideal\_points,sep=': ')

#Создание вспомогательной функции

f1 = -1\*funcs[0][0]\*x1- 1\*funcs[0][1]\*x2

f2 = -1\*funcs[1][0]\*x1 -1\*funcs[1][1]\*x2

f3 = -1\*funcs[2][0]\*x1 -1\*funcs[2][1]\*x2

fi = (f1-ideal\_points[0])\*\*2 + (f2 - ideal\_points[1])\*\*2 + (f3 - ideal\_points[2])\*\*2

fi1 = diff(fi,x1)

fi2 = diff(fi,x2)

k = 0

prev\_solution = 1

prev = fi.subs({x1:X[0,0],x2:X[1,0]})

print("ф({}):".format(k),prev)

output = []

while True:

output.append([X[0,0],X[1,0]])

x = X[0,0]

y = X[1,0]

aux = poly(fi1.subs({x1:x,x2:y})\*x1+ fi2.subs({x1:x,x2:y})\*x2)

res = linprog(c=[aux.coeffs()[0],aux.coeffs()[1]],A\_ub=d,b\_ub=d\_rhs)

Xn = np.matrix([res.x[0],res.x[1]]).transpose()

Xn = X + l\*(Xn-X)

fi\_Xn = fi.subs({x1:Xn[0,0],x2:Xn[1,0]})

solution = solve(diff(fi\_Xn,l))[0]

Xn = np.matrix([i[0,0].subs({l:solution}) for i in Xn]).transpose()

cur = fi.subs({x1:Xn[0,0],x2:Xn[1,0]})

if prev\_solution - solution > 0.01:

X = Xn

prev = cur

prev\_solution = solution

k+=1

else:

break

print("Ответ:")

print(f"X-{len(output)} = [{X[0,0]}, {X[1,0]}]")

print("f1", f1.subs({x1:X[0,0],x2:X[1,0]}),sep=":")

print("f2", f2.subs({x1:X[0,0],x2:X[1,0]}),sep=":")

print("f3", f3.subs({x1:X[0,0],x2:X[1,0]}),sep=":")

return output