

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И ПРОГРАММНЫХ ТЕХНОЛОГИЙ

Отчёт по лабораторной работе №1

Курс: «Методы оптимизации и принятия решений»

Тема: «Многокритериальная оптимизация»

Выполнил студент:

Бояркин Никита Сергеевич

Группа: 13541/3

Проверил:

Сиднев Александр Георгиевич

Санкт-Петербург
2018 г.

Содержание

1	Лабораторная работа №1	2
1.1	Цель работы	2
1.2	Программа работы	2
1.3	Индивидуальное задание	2
1.4	Ход работы	3
1.4.1	Обозначения	3
1.4.2	Критерии	3
1.4.3	Ограничения	3
1.4.4	Поиск оптимумов частных критериев	4
1.4.5	Аддитивная свертка критериев	5
1.4.6	Мультипликативная свертка критериев	7
1.4.7	Максимин или минимакс	8
1.4.8	Метод последовательных уступок	10
1.4.9	Метод достижения цели (fgoalattain)	14
1.4.10	Введение метрики в пространстве критериев	15
1.4.11	Оценка Парето-оптимальности полученных решений	17
1.4.12	Решение задачи стохастического программирования	17
1.5	Вывод	21

Лабораторная работа №1

1.1 Цель работы

Научиться решать задачи по многокритериальной оптимизации.

1.2 Программа работы

- Ввести обозначения.
- Формализовать критерии.
- Формализовать ограничения.
- Решить задачу следующими способами:
 - Поиск оптимальных частных критериев.
 - Аддитивной сверткой.
 - Мультипликативной сверткой.
 - Максимин или минимакс.
 - Методом последовательных уступок.
 - `fgoalattain1`.
 - Введением метрики в пространстве критериев.
- Решить задачу стохастического программирования.
- Написать вывод.

1.3 Индивидуальное задание

Задача 14

Компания *Nakia* выпускает под своим брендом телефоны трёх ценовых сегментов:

1. *LowEnd* – розничная цена аппарата 60\$, стоимость производства 30\$
2. *MiddleEnd* – розничная цена 300\$, стоимость производства 100\$
3. *HighEnd* – цена в розничной сети 2000\$, стоимость производства 220\$

Мощность фабрик компании такова, что достижимые объёмы выпуска: 70 млн дешевых устройств, 30 млн устройств среднего сегмента, 1 млн дорогих телефонов. Розничная сеть может продать не более 80 млн устройств в год.

Доход фирмы от продажи дополнений, можно заранее оценить по формуле $F = (0.1x_1 + 10x_2 + 70\sqrt{x_3})^{\frac{3}{2}}$.

При выпуске больших партий дешевых телефонов неизбежен антирекламный эффект, обусловленный относительно высоким процентом брака и падением престижа марки. Ущерб от антирекламы можно оценить по формуле $F = \ln(20x_1 + 3x_2 + 0.01x_3)$.

По каждому из сегментов компания должна производить не менее половины максимального объема выпуска.

Необходимо найти годовой объём производства телефонов каждого сегмента для достижения

1. Максимизации оборота

2. Минимизации затрат на производство
3. Максимизации средней цены телефона
4. Максимизации выручки от продажи дополнений
5. Минимизации антирекламного эффекта

1.4 Ход работы

1.4.1 Обозначения

Для решения задачи будем использовать следующие обозначения:

- x_1 – количество произведенных дешевых телефонов (в миллионах).
- x_2 – количество произведенных телефонов среднего сегмента (в миллионах).
- x_3 – количество произведенных дорогих телефонов (в миллионах).

1.4.2 Критерии

Введем следующие функции для определения критериев:

$$\begin{cases} F_1 = 60x_1 + 300x_2 + 2000x_3 \\ F_2 = (0.1x_1 + 10x_2 + 70\sqrt{x_3})^{\frac{3}{2}} \\ F_3 = 30x_1 + 100x_2 + 220x_3 \\ F_4 = \ln(20x_1 + 3x_2 + 0.01x_3) \\ F_5 = F_1/(x_1 + x_2 + x_3) \\ F_6 = (F_1 + F_2) - (F_3 + F_4) \end{cases}$$

Тогда задача сводится к минимизации или максимизации следующих функций:

1. Максимизация оборота – $F_6 \rightarrow \max$
2. Минимизация затрат на производство – $F_3 \rightarrow \min$
3. Максимизация средней цены телефона – $F_5 \rightarrow \max$
4. Максимизация выручки от продажи дополнений – $F_2 \rightarrow \max$
5. Минимизация антирекламного эффекта – $F_4 \rightarrow \min$

Значения взяты в миллионах для повышения точности вычислений в MATLAB, особенно для свертки. Кроме того, такое решение подразумевается исходя из задания.

* Функция F_1 используется только для вычисления критерия F_6 , поэтому не будет впоследствии рассчитываться отдельно.

1.4.3 Ограничения

Формализуем ограничения, приведенные в формулировке задания:

$$\begin{cases} x_1 \leq 70 \\ -x_1 \leq -35 \\ x_2 \leq 30 \\ -x_2 \leq -15 \\ x_3 \leq 1 \\ -x_3 \leq -0.5 \\ x_1 + x_2 + x_3 \leq 80 \end{cases}$$

1.4.4 Поиск оптимумов частных критериев

Разработаем программу для MATLAB, которая решает задачу, в соответствии с ограничениями, для каждого из критериев:

```

1 clear all;
2 close all;
3 clc;
4 format long g;
5
6 % -F1 -> min
7 mF1 = @(X) -(60 * X(1) + 300 * X(2) + 2000 * X(3));
8 % -F2 -> min
9 mF2 = @(X) -((0.1 * X(1) + 10 * X(2) + 70 * sqrt(X(3))) ^ 1.5);
10 % F3 -> min
11 pF3 = @(X) 30 * X(1) + 100 * X(2) + 220 * X(3);
12 % F4 -> min
13 pF4 = @(X) log(20 * X(1) + 3 * X(2) + 0.01 * X(3));
14 % -F5 -> min
15 mF5 = @(X) -((-mF1(X)) / (X(1) + X(2) + X(3)));
16 % -F6 -> min
17 mF6 = @(X) -(((mF1(X)) + (mF2(X))) - (pF3(X) + pF4(X)));
18
19 A = [1, 0, 0;
20      -1, 0, 0;
21      0, 1, 0;
22      0, -1, 0;
23      0, 0, 1;
24      0, 0, -1;
25      1, 1, 1];
26
27 B = [70;
28      -35;
29      30;
30      -15;
31      1;
32      -0.5;
33      80];
34
35 Aeq = [];
36 Beq = [];
37
38 S = [35; 15; 0.5];
39
40 [x1, result1] = fmincon(mF1, S, A, B, Aeq, Beq);
41 [x2, result2] = fmincon(mF2, S, A, B, Aeq, Beq);
42 [x3, result3] = fmincon(pF3, S, A, B, Aeq, Beq);
43 [x4, result4] = fmincon(pF4, S, A, B, Aeq, Beq);
44 [x5, result5] = fmincon(mF5, S, A, B, Aeq, Beq);
45 [x6, result6] = fmincon(mF6, S, A, B, Aeq, Beq);
46
47 formatter = '— F1 max —\n%s\n%s\n\n— F2 max —\n%s\n%s\n\n— F3 min —\n%s\n%s\n\n—
48             F4 min —\n%s\n%s\n\n— F5 max —\n%s\n%s\n\n— F6 max —\n%s\n%s\n\n';
49 s1 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', x1, sum(x1));
50 s2 = sprintf('!F1 = %.3f, F2 = %.3f, F3 = %.3f, F4 = %.3f, F5 = %.3f, F6 = %.3f', -
51             result1, -mF2(x1), pF3(x1), pF4(x1), -mF5(x1), -mF6(x1));
52 s3 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', x2, sum(x2));
53 s4 = sprintf('F1 = %.3f, !F2 = %.3f, F3 = %.3f, F4 = %.3f, F5 = %.3f, F6 = %.3f', -mF1(x2
54             ), -result2, pF3(x2), pF4(x2), -mF5(x2), -mF6(x2));
55 s5 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', x3, sum(x3));
56 s6 = sprintf('F1 = %.3f, F2 = %.3f, !F3 = %.3f, F4 = %.3f, F5 = %.3f, F6 = %.3f', -mF1(x3
57             ), -mF2(x3), result3, pF4(x3), -mF5(x3), -mF6(x3));
58 s7 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', x4, sum(x4));
59 s8 = sprintf('F1 = %.3f, F2 = %.3f, F3 = %.3f, !F4 = %.3f, F5 = %.3f, F6 = %.3f', -mF1(x4
60             ), -mF2(x4), pF3(x4), result4, -mF5(x4), -mF6(x4));
61 s9 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', x5, sum(x5));
62 s10 = sprintf('F1 = %.3f, F2 = %.3f, F3 = %.3f, F4 = %.3f, !F5 = %.3f, F6 = %.3f', -mF1(

```

```

x5), -mF2(x5), pF3(x5), pF4(x5), -result5, -mF6(x5));
58 s11 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', x6, sum(x6));
59 s12 = sprintf('F1 = %.3f, F2 = %.3f, F3 = %.3f, F4 = %.3f, F5 = %.3f, !F6 = %.3f', -mF1(
x6), -mF2(x6), pF3(x6), pF4(x6), -mF5(x6), -result6);
60 fprintf(formatter, s1, s2, s3, s4, s5, s6, s7, s8, s9, s10, s11, s12);

```

Результат решения задачи:

```

1 — F1 max —
2 x1 = 49.000, x2 = 30.000, x3 = 1.000, x1 + x2 + x3 = 80.000
3 !F1 = 13940.000, F2 = 7258.939, F3 = 4690.000, F4 = 6.975, F5 = 174.250, F6 = 16501.963
4
5 — F2 max —
6 x1 = 49.000, x2 = 30.000, x3 = 1.000, x1 + x2 + x3 = 80.000
7 F1 = 13939.995, !F2 = 7258.939, F3 = 4689.998, F4 = 6.975, F5 = 174.250, F6 = 16501.961
8
9 — F3 min —
10 x1 = 35.000, x2 = 15.000, x3 = 0.500, x1 + x2 + x3 = 50.500
11 F1 = 7600.000, F2 = 2892.251, !F3 = 2660.000, F4 = 6.613, F5 = 150.495, F6 = 7825.638
12
13 — F4 min —
14 x1 = 35.000, x2 = 15.000, x3 = 0.528, x1 + x2 + x3 = 50.528
15 F1 = 7656.563, F2 = 2921.812, F3 = 2666.229, !F4 = 6.613, F5 = 151.530, F6 = 7905.533
16
17 — F5 max —
18 x1 = 35.000, x2 = 30.000, x3 = 1.000, x1 + x2 + x3 = 66.000
19 F1 = 13100.000, F2 = 7218.316, F3 = 4270.000, F4 = 6.672, !F5 = 198.485, F6 = 16041.644
20
21 — F6 max —
22 x1 = 49.000, x2 = 30.000, x3 = 1.000, x1 + x2 + x3 = 80.000
23 F1 = 13940.000, F2 = 7258.939, F3 = 4690.000, F4 = 6.975, F5 = 174.250, !F6 = 16501.964

```

Результирующие значения в формате таблицы:

	$x_1, Mill$	$x_2, Mill$	$x_3, Mill$	$F_2, Mill\$$	$F_3, Mill\$$	$F_4, Mill\$$	$F_5, Mill\$$	$F_6, Mill\$$
F2 max	49	30	1	<u>7258.939</u>	4689.998	6.975	174.250	16501.961
F3 min	35	15	0.5	2892.251	<u>2660.000</u>	6.613	150.495	7825.638
F4 min	35	15	0.528	2921.812	2666.229	<u>6.613</u>	151.530	7905.533
F5 max	35	30	1	7218.316	4270.000	6.672	<u>198.485</u>	16041.644
F6 max	49	30	1	7258.939	4690.000	6.975	174.250	<u>16501.964</u>

* Функция F_1 используется только для вычисления критерия F_6 , поэтому не рассчитывается отдельно.

1.4.5 Аддитивная свертка критериев

Для использования метода аддитивной свертки необходимо выполнить нормировку критериев, с тем чтобы сделать их значения соизмеримыми, а единицы измерения безразмерными. Нормировка производится делением функции критерия на модуль ее минимума или максимума.

$$f_i(x) = F_i(x)/|F_i^{extr}|$$

Формула аддитивной свертки имеет вид:

$$F_a(x) = \sum_{i=1}^N \lambda_i f_i(x), 0 < \lambda_i < 1, \sum_i \lambda_i = 1,$$

где $f_i(x)$ - критерии оптимальности, N - их общее число, а λ_i - коэффициенты важности. Примем коэффициенты важности равными $\lambda_2 = 0.15, \lambda_3 = 0.15, \lambda_4 = 0.05, \lambda_5 = 0.05, \lambda_6 = 0.6$. Коэффициент при F_6 очевидно наибольший, так как итоговый оборот интересует прежде всего.

```

1 clear all;
2 close all;
3 clc;
4 format long g;

```

```

5
6 % -F1 -> min
7 mF1 = @(X) -(60 * X(1) + 300 * X(2) + 2000 * X(3));
8 % -F2 -> min
9 mF2 = @(X) -((0.1 * X(1) + 10 * X(2) + 70 * sqrt(X(3))) ^ 1.5);
10 % F3 -> min
11 pF3 = @(X) 30 * X(1) + 100 * X(2) + 220 * X(3);
12 % F4 -> min
13 pF4 = @(X) log(20 * X(1) + 3 * X(2) + 0.01 * X(3));
14 % -F5 -> min
15 mF5 = @(X) -((-mF1(X)) / (X(1) + X(2) + X(3)));
16 % -F6 -> min
17 mF6 = @(X) -(((-mF1(X)) + (-mF2(X))) - (pF3(X) + pF4(X)));
18
19 cF2 = 0.15;
20 rF2 = 7258.939;
21 cF3 = 0.15;
22 rF3 = 2660;
23 cF4 = 0.05;
24 rF4 = 6.613;
25 cF5 = 0.05;
26 rF5 = 198.485;
27 cF6 = 0.6;
28 rF6 = 16501.964;
29
30 pFS = @(X) cF2 * (mF2(X) / rF2) + cF3 * (pF3(X) / rF3) + cF4 * (pF4(X) / rF4) + cF5 * (
    mF5(X) / rF5) + cF6 * (mF6(X) / rF6);
31
32 A = [1, 0, 0;
33      -1, 0, 0;
34       0, 1, 0;
35       0, -1, 0;
36       0, 0, 1;
37       0, 0, -1;
38       1, 1, 1];
39
40 B = [70;
41      -35;
42       30;
43      -15;
44       1;
45      -0.5;
46       80];
47
48 Aeq = [];
49 Beq = [];
50
51 S = [35; 15; 0.5];
52
53 [xS, resultS] = fmincon(pFS, S, A, B, Aeq, Beq);
54
55 formatter = '— FS —\n%s\n%s\n%s\n%s\n\n';
56 s1 = sprintf('FS = %.3f', resultS);
57 s2 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', xS, sum(xS));
58 s3 = sprintf('F2 = %.3f, F3 = %.3f, F4 = %.3f, F5 = %.3f, F6 = %.3f', -mF2(xS), pF3(xS),
    pF4(xS), -mF5(xS), -mF6(xS));
59 s4 = sprintf('F2/rF2 = %.3f%%, F3/rF3 = %.3f%%, F4/rF4 = %.3f%%, F5/rF5 = %.3f%%, F6/rF6
    = %.3f%%', -mF2(xS) / rF2 * 100, pF3(xS) / rF3 * 100, pF4(xS) / rF4 * 100, -mF5(xS) /
    rF5 * 100, -mF6(xS) / rF6 * 100);
60 fprintf(formatter, s1, s2, s3, s4);

```

Результат решения при помощи аддитивной свертки:

```

1 — FS —
2 FS = -0.491
3 x1 = 35.000, x2 = 30.000, x3 = 1.000, x1 + x2 + x3 = 66.000
4 F2 = 7218.316, F3 = 4270.000, F4 = 6.672, F5 = 198.485, F6 = 16041.644

```

5 F2/rF2 = 99.440%, F3/rF3 = 160.526%, F4/rF4 = 100.893%, F5/rF5 = 100.000%, F6/rF6 = 97.211%

Стоит отметить, что при больших значениях минимума или максимума функций критерии оптимальности становятся очень маленькими, и поэтому MATLAB выдает не совсем корректные результаты. Это еще раз подтверждает правильность взятия значений x_1, x_2, x_3 в миллионах.

	result, Mill\$	proportion, %	difference, %
F2 max	7218.316	99.44	0.56
F3 min	4270.000	160.526	60.526
F4 min	6.672	100.893	0.893
F5 max	174.25	100	0
F6 max	16501.96	97.211	2.789
Mean			12.95

* Функция F_1 используется только для вычисления критерия F_6 , поэтому не рассчитывается отдельно.

1.4.6 Мультипликативная свертка критериев

Формула мультипликативной свертки имеет вид:

$$F_m(x) = \prod_{i=1}^N f_i(x)^{\lambda_i}, 0 < \lambda_i < 1, \sum_i \lambda_i = 1,$$

где $f_i(x)$ - критерии оптимальности, N - их общее число, а λ_i - коэффициенты важности. Коэффициенты важности оставим равными $\lambda_2 = 0.15, \lambda_3 = 0.15, \lambda_4 = 0.05, \lambda_5 = 0.05, \lambda_6 = 0.6$.

```

1 clear all;
2 close all;
3 clc;
4 format long g;
5
6 % 1 / F1 -> min
7 mF1 = @(X) 1 / (60 * X(1) + 300 * X(2) + 2000 * X(3));
8 % 1 / F2 -> min
9 mF2 = @(X) 1 / ((0.1 * X(1) + 10 * X(2) + 70 * sqrt(X(3))) ^ 1.5);
10 % F3 -> min
11 pF3 = @(X) 30 * X(1) + 100 * X(2) + 220 * X(3);
12 % F4 -> min
13 pF4 = @(X) log(20 * X(1) + 3 * X(2) + 0.01 * X(3));
14 % 1 / F5 -> min
15 mF5 = @(X) 1 / ((1 / mF1(X)) / (X(1) + X(2) + X(3)));
16 % 1 / F6 -> min
17 mF6 = @(X) 1 / (((1 / mF1(X)) + (1 / mF2(X))) - (pF3(X) + pF4(X)));
18
19 cF2 = 0.15;
20 rF2 = 7258.939;
21 cF3 = 0.15;
22 rF3 = 2660;
23 cF4 = 0.05;
24 rF4 = 6.613;
25 cF5 = 0.05;
26 rF5 = 198.485;
27 cF6 = 0.6;
28 rF6 = 16501.964;
29
30 pFS = @(X) nthroot(mF2(X) * rF2, 1 / cF2) * nthroot(pF3(X) / rF3, 1 / cF3) * nthroot(pF4(X) / rF4, 1 / cF4) * nthroot(mF5(X) * rF5, 1 / cF5) * nthroot(mF6(X) * rF6, 1 / cF6);
31
32 A = [1, 0, 0;
33      -1, 0, 0;
34       0, 1, 0;
35       0, -1, 0;
36       0, 0, 1;

```



```

37     0, 0, -1;
38     1, 1, 1];
39
40 B = [70;
41     -35;
42     30;
43     -15;
44     1;
45     -0.5;
46     80];
47
48 Aeq = [];
49 Beq = [];
50
51 S = [35; 15; 0.5];
52
53 [xS, resultS] = fmincon(pFS, S, A, B, Aeq, Beq);
54
55 formatter = '— FS —\n%s\n%s\n%s\n%s\n\n';
56 s1 = sprintf('FS = %.3f', resultS);
57 s2 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', xS, sum(xS));
58 s3 = sprintf('F2 = %.3f, F3 = %.3f, F4 = %.3f, F5 = %.3f, F6 = %.3f', 1 / mF2(xS), pF3(xS), pF4(xS), 1 / mF5(xS), 1 / mF6(xS));
59 s4 = sprintf('F2/rF2 = %.3f%%, F3/rF3 = %.3f%%, F4/rF4 = %.3f%%, F5/rF5 = %.3f%%, F6/rF6 = %.3f%%', (1 / mF2(xS)) / rF2 * 100, pF3(xS) / rF3 * 100, pF4(xS) / rF4 * 100, (1 / mF5(xS)) / rF5 * 100, (1 / mF6(xS)) / rF6 * 100);
60 fprintf(formatter, s1, s2, s3, s4);

```

Результат решения при помощи мультипликативной свертки:

```

1 — FS —
2 FS = 1.093
3 x1 = 35.000, x2 = 30.000, x3 = 1.000, x1 + x2 + x3 = 66.000
4 F2 = 7218.315, F3 = 4270.004, F4 = 6.672, F5 = 198.485, F6 = 16041.646
5 F2/rF2 = 99.440%, F3/rF3 = 160.526%, F4/rF4 = 100.893%, F5/rF5 = 100.000%, F6/rF6 = 97.211%

```

По результирующему значению мультипликативной свертки можно заметить, что она лучше справляется в ситуации с очень маленькими критериями оптимальности, чем адаптивная свертка.

	result, Mill\$	proportion, %	difference, %
F2 max	7218.315	99.44	0.56
F3 min	4270.004	160.526	60.526
F4 min	6.672	100.893	0.893
F5 max	198.485	100	0
F6 max	16041.646	97.211	2.789
Mean			12.95

Результат аналогичен аддитивной свертке.

* Функция F_1 используется только для вычисления критерия F_6 , поэтому не рассчитывается отдельно.

1.4.7 Максимин или минимакс

Максиминную свертку представим в следующем виде:

$$C_i(a) = \min w_i C_i(a)$$

Решение a^* является наилучшим, если для всех a выполняется условие:

$$C(a^*) \geq C(a)$$

или

$$a^* = \operatorname{argmax} C(a) = \operatorname{argmax} \min w_i C_i(a)$$

Решение задачи в среде MATLAB:

```

1 clear all;
2 close all;
3 clc;
4 format long g;
5
6 % 1 / F1 -> min
7 mF1 = @(X) 1 / (60 * X(1) + 300 * X(2) + 2000 * X(3));
8 % 1 / F2 -> min
9 mF2 = @(X) 1 / ((0.1 * X(1) + 10 * X(2) + 70 * sqrt(X(3))) ^ 1.5);
10 % F3 -> min
11 pF3 = @(X) 30 * X(1) + 100 * X(2) + 220 * X(3);
12 % F4 -> min
13 pF4 = @(X) log(20 * X(1) + 3 * X(2) + 0.01 * X(3));
14 % 1 / F5 -> min
15 mF5 = @(X) 1 / ((1 / mF1(X)) / (X(1) + X(2) + X(3)));
16 % 1 / F6 -> min
17 mF6 = @(X) 1 / (((1 / mF1(X)) + (1 / mF2(X))) - (pF3(X) + pF4(X)));
18
19 rF1 = 13940;
20 rF2 = 7258.939;
21 rF3 = 2660;
22 rF4 = 6.613;
23 rF5 = 198.485;
24 rF6 = 16501.964;
25
26 A = [1, 0, 0;
27      -1, 0, 0;
28       0, 1, 0;
29       0, -1, 0;
30       0, 0, 1;
31       0, 0, -1;
32       1, 1, 1];
33
34 B = [70;
35      -35;
36       30;
37      -15;
38       1;
39      -0.5;
40      80];
41
42 Aeq = [];
43 Beq = [];
44
45 S = [35; 15; 0.5];
46
47 [xS, resultS] = fminimax(@m4f, S, A, B, Aeq, Beq);
48
49 formatter = '--- FS ---\n%s\n%s\n%s\n%s\n\n';
50 s1 = sprintf('FS = (%.3f, %.3f, %.3f, %.3f, %.3f)', resultS(2:6));
51 s2 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', xS, sum(xS));
52 s3 = sprintf('F2 = %.3f, F3 = %.3f, F4 = %.3f, F5 = %.3f, F6 = %.3f', 1 / mF2(xS), pF3(xS), pF4(xS), 1 / mF5(xS), 1 / mF6(xS));
53 s4 = sprintf('F2/rF2 = %.3f%%, F3/rF3 = %.3f%%, F4/rF4 = %.3f%%, F5/rF5 = %.3f%%, F6/rF6 = %.3f%%', (1 / mF2(xS)) / rF2 * 100, pF3(xS) / rF3 * 100, pF4(xS) / rF4 * 100, (1 / mF5(xS)) / rF5 * 100, (1 / mF6(xS)) / rF6 * 100);
54 fprintf(formatter, s1, s2, s3, s4);

```

```

1 function result = m4f(X)
2     % 1 / F1 -> min
3     mF1 = 1 / (60 * X(1) + 300 * X(2) + 2000 * X(3));
4     % 1 / F2 -> min
5     mF2 = 1 / ((0.1 * X(1) + 10 * X(2) + 70 * sqrt(X(3))) ^ 1.5);
6     % F3 -> min
7     pF3 = 30 * X(1) + 100 * X(2) + 220 * X(3);

```

```

8      % F4 -> min
9      pF4 = log(20 * X(1) + 3 * X(2) + 0.01 * X(3));
10     % 1 / F5 -> min
11     mF5 = 1 / ((1 / mF1) / (X(1) + X(2) + X(3)));
12     % 1 / F6 -> min
13     mF6 = 1 / (((1 / mF1) + (1 / mF2)) - (pF3 + pF4));
14
15     rF1 = 13940;
16     rF2 = 7258.939;
17     rF3 = 2660;
18     rF4 = 6.613;
19     rF5 = 198.485;
20     rF6 = 16501.964;
21
22     result(1) = mF1 * rF1;
23     result(2) = mF2 * rF2;
24     result(3) = pF3 / rF3;
25     result(4) = pF4 / rF4;
26     result(5) = mF5 * rF5;
27     result(6) = mF6 * rF6;
28 end

```

Результат решения задачи:

```

1  — FS —
2  FS = (1.353, 1.353, 1.005, 1.061, 1.285)
3  x1 = 35.000, x2 = 23.294, x3 = 1.000, x1 + x2 + x3 = 59.294
4  F2 = 5364.414, F3 = 3599.419, F4 = 6.646, F5 = 187.004, F6 = 12846.607
5  F2/rF2 = 73.901%, F3/rF3 = 135.317%, F4/rF4 = 100.503%, F5/rF5 = 94.216%, F6/rF6 =
    77.849%

```

Результат решения задачи в виде таблицы:

	result, Mill\$	proportion, %	difference, %
F2 max	5364.414	73.901	26.099
F3 min	3599.419	135.317	35.317
F4 min	6.646	100.503	0.503
F5 max	187.004	94.216	5.784
F6 max	12846.607	77.849	22.151
Mean			17.97

* Функция F_1 используется только для вычисления критерия F_6 , поэтому не рассчитывается отдельно.

1.4.8 Метод последовательных уступок

Для метода последовательных уступок данные целевые функции не очень подходят, ввиду их нелинейности. Введем три новых критерия, которые хорошо иллюстрируют метод последовательных уступок:

$$\begin{cases} F_7 = 30x_1 + 200x_2 + 1780x_3 \\ F_8 = x_1 + 10x_3 \\ F_9 = x_1 + 2x_2 \end{cases}$$

Тогда задача сводится к минимизации или максимизации следующих функций:

1. Максимизация оборота (без антирекламного эффекта и продажи дополнений) – $F_7 \rightarrow \max$
2. Максимизация доли дешевых и дорогих телефонов – $F_8 \rightarrow \max$
3. Минимизация доли средних и дешевых телефонов – $F_9 \rightarrow \min$

Расположим критерии в порядке значимости:

$$F_7 > F_8 > F_9$$

Для решения задачи была выбрана уступка равная 15%.

Решение задачи для критерия F_7 :

```

1 clear all;
2 close all;
3 clc;
4 format long g;
5
6 % -F7 -> min
7 mF7 = @(X) -(30 * X(1) + 200 * X(2) + 1780 * X(3));
8 % -F8 -> min
9 mF8 = @(X) -(X(1) + 10 * X(3));
10 % F9 -> min
11 pF9 = @(X) X(1) + 2 * X(2);
12
13 A = [1, 0, 0;
14      -1, 0, 0;
15       0, 1, 0;
16       0, -1, 0;
17       0, 0, 1;
18       0, 0, -1;
19       1, 1, 1];
20
21 B = [70;
22      -35;
23       30;
24      -15;
25       1;
26      -0.5;
27       80];
28
29 Aeq = [];
30 Beq = [];
31
32 S = [35; 15; 0.5];
33
34 [x7, result7] = fmincon(mF7, S, A, B, Aeq, Beq);
35
36 formatter = '--- F7 max ---\n%s\n%s\n\n';
37 s1 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', x7, sum(x7));
38 s2 = sprintf('!F7 = %.3f, F8 = %.3f, F9 = %.3f', -result7, -mF8(x7), pF9(x7));
39 fprintf(formatter, s1, s2);

```

Результат решения задачи для критерия F_7 :

```

1 --- F7 max ---
2 x1 = 49.000, x2 = 30.000, x3 = 1.000, x1 + x2 + x3 = 80.000
3 !F7 = 9250.000, F8 = 59.000, F9 = 109.000

```

Результаты решения для критерия F_7 в виде таблицы:

	result, Mill\$	proportion, %	difference, %
F7	9250	100	0
F8	59	-	-
F9	109	-	-

Максимум целевой функции F_7 равен 9250. Для расчета максимума F_8 будет добавлено новое ограничение:

$$F_7 \leq 9250 * 0.85$$

$$F_7 \leq 7862.5$$

Решение задачи для критерия F_8 :

```

1 clear all;
2 close all;
3 clc;
4 format long g;

```

```

5
6 % -F7 -> min
7 mF7 = @(X) -(30 * X(1) + 200 * X(2) + 1780 * X(3));
8 % -F8 -> min
9 mF8 = @(X) -(X(1) + 10 * X(3));
10 % F9 -> min
11 pF9 = @(X) X(1) + 2 * X(2);
12
13 K = 0.85
14
15 rF7 = 9250;
16
17 A = [1, 0, 0;
18      -1, 0, 0;
19      0, 1, 0;
20      0, -1, 0;
21      0, 0, 1;
22      0, 0, -1;
23      1, 1, 1;
24      -30, -200, -1780];
25
26 B = [70;
27      -35;
28      30;
29      -15;
30      1;
31      -0.5;
32      80;
33      -rF7 * K];
34
35 Aeq = [];
36 Beq = [];
37
38 S = [35; 15; 0.5];
39
40 [x8, result8] = fmincon(mF8, S, A, B, Aeq, Beq);
41
42 fprintf('%.3f >= F7 >= %.1f * %.3f\n', rF7, K, rF7);
43 fprintf('%.3f >= F7 >= %.3f\n\n', rF7, rF7 * K);
44
45 formatter = '— F8 max —\n%s\n%s\n\n';
46 s1 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', x8, sum(x8));
47 s2 = sprintf('F7 = %.3f, !F8 = %.3f, F9 = %.3f', -mF7(x8), -result8, pF9(x8));
48 fprintf(formatter, s1, s2);

```

Результат решения задачи для критерия F_8 :

```

1 9250.000 >= F7 >= 0.8 * 9250.000
2 9250.000 >= F7 >= 7862.500
3
4 — F8 max —
5 x1 = 57.162, x2 = 21.838, x3 = 1.000, x1 + x2 + x3 = 80.000
6 F7 = 7862.500, !F8 = 67.162, F9 = 100.838

```

Результаты решения для критерия F_8 в виде таблицы:

	result, Mill\$	proportion, %	difference, %
F7	7862.5	85	15
F8	67.162	100	0
F9	100.838	-	-

Максимум целевой функции F_8 равен 67.162. Для расчета максимума F_9 будет добавлено новое ограничение:

$$F_8 \leq 67.162 * 0.85$$

$$F_8 \leq 57.088$$

Решение задачи для критерия F_9 :

```

1 clear all;
2 close all;
3 clc;
4 format long g;
5
6 % -F7 -> min
7 mF7 = @(X) -(30 * X(1) + 200 * X(2) + 1780 * X(3));
8 % -F8 -> min
9 mF8 = @(X) -(X(1) + 10 * X(3));
10 % F9 -> min
11 pF9 = @(X) X(1) + 2 * X(2);
12
13 K = 0.85
14
15 rF7 = 9250;
16 rF8 = 67.162;
17
18 A = [1, 0, 0;
19      -1, 0, 0;
20       0, 1, 0;
21       0, -1, 0;
22       0, 0, 1;
23       0, 0, -1;
24       1, 1, 1;
25      -30, -200, -1780;
26      -1, 0, -10];
27
28 B = [70;
29      -35;
30       30;
31      -15;
32       1;
33      -0.5;
34       80;
35      -rF7 * K;
36      -rF8 * K];
37
38 Aeq = [];
39 Beq = [];
40
41 S = [35; 15; 0.5];
42
43 [x9, result9] = fmincon(pF9, S, A, B, Aeq, Beq);
44
45 fprintf('%.3f >= F7 >= %.2f * %.3f\n', rF7, K, rF7);
46 fprintf('%.3f >= F7 >= %.3f\n\n', rF7, rF7 * K);
47
48 fprintf('%.3f >= F8 >= %.2f * %.3f\n', rF8, K, rF8);
49 fprintf('%.3f >= F8 >= %.3f\n\n', rF8, rF8 * K);
50
51 formatter = '— F9 max —\n%s\n%s\n\n';
52 s1 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', x9, sum(x9));
53 s2 = sprintf('F7 = %.3f, F8 = %.3f, !F9 = %.3f', -mF7(x9), -mF8(x9), result9);
54 fprintf(formatter, s1, s2);

```

Результат решения задачи для критерия F_9 :

```

1 9250.000 >= F7 >= 0.85 * 9250.000
2 9250.000 >= F7 >= 7862.500
3
4 67.162 >= F8 >= 0.85 * 67.162
5 67.162 >= F8 >= 57.088
6

```

```

7 — F9 max —
8 x1 = 47.088, x2 = 23.349, x3 = 1.000, x1 + x2 + x3 = 71.437
9 F7 = 7862.500, F8 = 57.088, !F9 = 93.786

```

Результаты решения для критерия F_9 в виде таблицы:

	result, Mill\$	proportion, %	difference, %
F7	7862.5	85	15
F8	57.088	85	15
F9	93.786	100	0

Отличие результатов целевых функций от максимальных или минимальных значений не превышают принятое значение уступки 15%.

1.4.9 Метод достижения цели (fgoalattain)

Функция fgoalattain решает задачу достижения цели, которая является одной из формулировок задач для векторной оптимизации. Аргументы fgoalattain схожи с функцией fgoalattain, за исключением добавления целевых значений и весов. Кроме того, одновременно ищутся оптимальные значения для всех целевых функций, а не для одной.

Решение задачи в среде MATLAB:

```

1 clear all;
2 close all;
3 clc;
4 format long g;
5
6 % -F1 -> min
7 mF1 = @(X) -(60 * X(1) + 300 * X(2) + 2000 * X(3));
8 % -F2 -> min
9 mF2 = @(X) -((0.1 * X(1) + 10 * X(2) + 70 * sqrt(X(3))) ^ 1.5);
10 % F3 -> min
11 pF3 = @(X) 30 * X(1) + 100 * X(2) + 220 * X(3);
12 % F4 -> min
13 pF4 = @(X) log(20 * X(1) + 3 * X(2) + 0.01 * X(3));
14 % -F5 -> min
15 mF5 = @(X) -((-mF1(X)) / (X(1) + X(2) + X(3)));
16 % -F6 -> min
17 mF6 = @(X) -((( -mF1(X)) + (-mF2(X))) - (pF3(X) + pF4(X)));
18
19 rF1 = -13940;
20 rF2 = -7258.939;
21 rF3 = 2660;
22 rF4 = 6.613;
23 rF5 = -198.485;
24 rF6 = -16501.964;
25
26 pFS = @(X) [mF2(X), pF3(X), pF4(X), mF5(X), mF6(X)];
27
28 G = [rF2, rF3, rF4, rF5, rF6];
29
30 W = abs(G);
31
32 A = [1, 0, 0;
33      -1, 0, 0;
34       0, 1, 0;
35       0, -1, 0;
36       0, 0, 1;
37       0, 0, -1;
38       1, 1, 1];
39
40 B = [70;
41      -35;
42       30;
43      -15;

```

```

44     1;
45     -0.5;
46     80];
47
48 Aeq = [];
49 Beq = [];
50
51 S = [35; 15; 0.5];
52
53 [xS, resultS] = fgoalattain(pFS, S, G, W, A, B, Aeq, Beq);
54
55 formatter = '— FS —\n%s\n%s\n%s\n%s\n\n';
56 s1 = sprintf('FS = (%.3f, %.3f, %.3f, %.3f, %.3f)', resultS);
57 s2 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', xS, sum(xS));
58 s3 = sprintf('F2 = %.3f, F3 = %.3f, F4 = %.3f, F5 = %.3f, F6 = %.3f', -mF2(xS), pF3(xS),
59             pF4(xS), -mF5(xS), -mF6(xS));
60 s4 = sprintf('F2/rF2 = %.3f%%, F3/rF3 = %.3f%%, F4/rF4 = %.3f%%, F5/rF5 = %.3f%%, F6/rF6 = %.3f%%',
61             mF2(xS) / rF2 * 100, pF3(xS) / rF3 * 100, pF4(xS) / rF4 * 100, mF5(xS) /
62             rF5 * 100, mF6(xS) / rF6 * 100);
63 fprintf(formatter, s1, s2, s3, s4);

```

Результат решения задачи:

```

1 — FS —
2 FS = (-5038.049, 3473.833, 6.641, -184.559, -12269.075)
3 x1 = 35.000, x2 = 22.038, x3 = 1.000, x1 + x2 + x3 = 58.038
4 F2 = 5038.049, F3 = 3473.833, F4 = 6.641, F5 = 184.559, F6 = 12269.075
5 F2/rF2 = 69.405%, F3/rF3 = 130.595%, F4/rF4 = 100.429%, F5/rF5 = 92.984%, F6/rF6 =
  74.349%

```

Результаты решения для критерия в виде таблицы:

	result, Mill\$	proportion, %	difference, %
F2 max	5038.049	69.405	30.595
F3 min	3473.833	130.595	30.595
F4 min	6.641	100.429	0.429
F5 max	184.559	92.984	7.016
F6 max	12269.075	74.349	25.651
Mean			18.86

* Функция F_1 используется только для вычисления критерия F_6 , поэтому не рассчитывается отдельно.

1.4.10 Введение метрики в пространстве критериев

Для перехода к однокритериальной задаче оптимизации методом введения метрики в пространстве целевых функций необходимо определить координаты идеальной точки $a_i = (f_1^*, f_2^*, \dots, f_1^N)$, где $f_i = \min(f_i(x))$. Данные оптимальные значения уже известны из предыдущих пунктов работы и равняются:

$$a = [7258.939, 2660, 6.613, 198.485, 16501.964]$$

Введем в пространстве критериев метрику в виде евклидова расстояния:

$$p(y, a) = \left(\sum_{i=1}^N (a_i - y_i)^2 \right)^{\frac{1}{2}}$$

Тогда за целевую функцию (обобщенный критерий), с учётом необходимости нормировки, можно взять выражение:

$$f = \sum_{i=1}^N \left(\frac{a_i - f_i}{f_i^*} \right)^2 = \sum_{i=1}^N \left(1 - \frac{f_i}{f_i^*} \right)^2$$

Решение задачи в среде MATLAB:


```

1 clear all;
2 close all;
3 clc;
4 format long g;
5
6 % -F1 -> min
7 mF1 = @(X) -(60 * X(1) + 300 * X(2) + 2000 * X(3));
8 % -F2 -> min
9 mF2 = @(X) -((0.1 * X(1) + 10 * X(2) + 70 * sqrt(X(3))) ^ 1.5);
10 % F3 -> min
11 pF3 = @(X) 30 * X(1) + 100 * X(2) + 220 * X(3);
12 % F4 -> min
13 pF4 = @(X) log(20 * X(1) + 3 * X(2) + 0.01 * X(3));
14 % -F5 -> min
15 mF5 = @(X) -((-mF1(X)) / (X(1) + X(2) + X(3)));
16 % -F6 -> min
17 mF6 = @(X) -(((-mF1(X)) + (-mF2(X))) - (pF3(X) + pF4(X)));
18
19 rF1 = -13940;
20 rF2 = -7258.939;
21 rF3 = 2660;
22 rF4 = 6.613;
23 rF5 = -198.485;
24 rF6 = -16501.964;
25
26 pFS = @(X) (1 - mF2(X) / rF2) ^ 2 + (1 - pF3(X) / rF3) ^ 2 + (1 - pF4(X) / rF4) ^ 2 + (1
    - mF5(X) / rF5) ^ 2 + (1 - mF6(X) / rF6) ^ 2;
27
28 A = [1, 0, 0;
29      -1, 0, 0;
30       0, 1, 0;
31       0, -1, 0;
32       0, 0, 1;
33       0, 0, -1;
34       1, 1, 1];
35
36 B = [70;
37      -35;
38       30;
39      -15;
40       1;
41      -0.5;
42       80];
43
44 Aeq = [];
45 Beq = [];
46
47 S = [35; 15; 0.5];
48
49 [xS, resultS] = fmincon(pFS, S, A, B, Aeq, Beq);
50
51 formatter = '— FS —\n%s\n%s\n%s\n%s\n\n';
52 s1 = sprintf('FS = %.3f', resultS);
53 s2 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', xS, sum(xS));
54 s3 = sprintf('F2 = %.3f, F3 = %.3f, F4 = %.3f, F5 = %.3f, F6 = %.3f', -mF2(xS), pF3(xS),
    pF4(xS), -mF5(xS), -mF6(xS));
55 s4 = sprintf('F2/rF2 = %.3f%%, F3/rF3 = %.3f%%, F4/rF4 = %.3f%%, F5/rF5 = %.3f%%, F6/rF6
    = %.3f%%', mF2(xS) / rF2 * 100, pF3(xS) / rF3 * 100, pF4(xS) / rF4 * 100, mF5(xS) /
    rF5 * 100, mF6(xS) / rF6 * 100);
56 fprintf(formatter, s1, s2, s3, s4);

```

Результат решения задачи:

```

1 — FS —
2 FS = 0.243
3 x1 = 35.000, x2 = 24.145, x3 = 1.000, x1 + x2 + x3 = 60.145

```

⁴ $F_2 = 5589.230$, $F_3 = 3684.450$, $F_4 = 6.650$, $F_5 = 188.602$, $F_6 = 13241.479$
⁵ $F_2/rF_2 = 76.998\%$, $F_3/rF_3 = 138.513\%$, $F_4/rF_4 = 100.553\%$, $F_5/rF_5 = 95.021\%$, $F_6/rF_6 = 80.242\%$

Результаты решения для критерия в виде таблицы:

	result, Mill\$	proportion, %	difference, %
F2 max	5589.230	76.998	23.002
F3 min	3684.450	138.513	38.513
F4 min	6.650	100.553	0.553
F5 max	188.602	95.021	4.979
F6 max	13241.479	80.242	19.758
Mean			17.361

* Функция F_1 используется только для вычисления критерия F_6 , поэтому не рассчитывается отдельно.

1.4.11 Оценка Парето-оптимальности полученных решений

Выделим результаты решения задачи различными методами в отдельную таблицу. Метод последовательных уступок из-за нелинейности некоторых целевых функций не попадает в таблицу.

Метод	$x_1, Mill$	$x_2, Mill$	$x_3, Mill$	$F_2, Mill\$$	$F_3, Mill\$$	$F_4, Mill\$$	$F_5, Mill\$$	$F_6, Mill\$$	Средняя разница %
Аддитивная свертка	35	30	1	7218.316	4270.000	6.672	198.485	16041.644	12.95
Мультипликативная свертка	35	30	1	7218.315	4270.004	6.672	198.485	16041.646	12.95
Минимакс	35	23.294	1	5364.414	3599.419	6.646	187.004	12846.607	17.97
Метод последовательных уступок	-	-	-	-	-	-	-	-	-
Метод достижения цели	35	22.038	1	5038.049	3473.833	6.641	184.559	12269.075	18.86
Введение метрики в пространстве критериев	35	24.145	1	5589.230	3684.450	6.650	188.602	13241.479	17.361

* Функция F_1 используется только для вычисления критерия F_6 , поэтому не рассчитывается отдельно.

Парето-оптимальными по соотношению доходов к расходам можно назвать только аддитивную и мультипликативную свертку, так как при их использовании получается наибольший оборот (F_6), а оборот как раз так и характеризует разницу между доходами и расходами. Для этих методов наибольший оборот получается благодаря введению коэффициентов значимости.

1.4.12 Решение задачи стохастического программирования

Рассмотрим задачу стохастического программирования на основе задачи однокритериальной оптимизации, которая была получена из исходной методом введения метрики в пространстве критериев.

Преобразуем последнее ограничение системы:

$$x_1 + x_2 + x_3 \leq 80$$

в вероятностное, тогда:

$$P\{\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 \leq 80\} \geq \alpha$$

где все a_i нормально распределены и имеют следующие математические ожидания и дисперсии:

$$M[\alpha_1] = 1, M[\alpha_2] = 1, M[\alpha_3] = 1$$

$$\sigma[\alpha_1] = 0.5, \sigma[\alpha_2] = 0.5, \sigma[\alpha_3] = 0.5$$

где СКО равняется половине математического ожидания. По таблице функции нормального распределения находим коэффициенты K_α :

$$K_{0.5} = 0, K_{0.6} = 0.2533, K_{0.7} = 0.5244, K_{0.8} = 0.8416, K_{0.9} = 1.2816$$

Таким образом, вероятностное ограничение становится эквивалентно детерминированному неравенству:

$$x_1 + x_2 + x_3 + K_\alpha \sqrt{0.5x_1^2 + 0.5x_2^2 + 0.5x_3^2} \leq 80$$

Решение задачи в среде MATLAB:

```

1 clear all;
2 close all;
3 clc;
4 format long g;
5
6 % -F1 -> min
7 mF1 = @(X) -(60 * X(1) + 300 * X(2) + 2000 * X(3));
8 % -F2 -> min
9 mF2 = @(X) -((0.1 * X(1) + 10 * X(2) + 70 * sqrt(X(3))) ^ 1.5);
10 % F3 -> min
11 pF3 = @(X) 30 * X(1) + 100 * X(2) + 220 * X(3);
12 % F4 -> min
13 pF4 = @(X) log(20 * X(1) + 3 * X(2) + 0.01 * X(3));
14 % -F5 -> min
15 mF5 = @(X) -((-mF1(X)) / (X(1) + X(2) + X(3)));
16 % -F6 -> min
17 mF6 = @(X) -((-mF1(X)) + (-mF2(X))) - (pF3(X) + pF4(X));
18
19 rF1 = -13940;
20 rF2 = -7258.939;
21 rF3 = 2660;
22 rF4 = 6.613;
23 rF5 = -198.485;
24 rF6 = -16501.964;
25
26 pFS = @(X) (1 - mF2(X) / rF2) ^ 2 + (1 - pF3(X) / rF3) ^ 2 + (1 - pF4(X) / rF4) ^ 2 + (1 - mF5(X) / rF5) ^ 2 + (1 - mF6(X) / rF6) ^ 2;
27
28 A = [1, 0, 0;
29      -1, 0, 0;
30       0, 1, 0;
31       0, -1, 0;
32       0, 0, 1;
33       0, 0, -1;
34       1, 1, 1];
35
36 B = [70;
37      -35;
38       30;
39      -15;
40       1;
41      -0.5;
42       80];
43
44 Aeq = [];
45 Beq = [];
46
47 lb = [];
48 ub = [];
49

```

```

50 S = [35; 15; 0.5];
51
52 O = optimoptions('fmincon', 'Display', 'none');
53
54 [xS, resultS] = fmincon(pFS, S, A, B, Aeq, Beq, lb, ub, [], O);
55
56 global K;
57
58 Ka = icdf('Normal', 0.1 : 0.1 : 0.9, 0, 1);
59 sizeK = size(Ka, 2);
60 xV = zeros(3, sizeK);
61 rV = zeros(3, sizeK);
62 for i = 1 : 1 : sizeK
63     K = Ka(i);
64     fmincon(pFS, S, A, B, Aeq, Beq);
65
66     [xV(:, i), rV(:, i)] = fmincon(pFS, S, A(1:6, :), B(1:6, :), Aeq, Beq, lb, ub, @m8f,
67     O);
68 end
69
70 mean = (abs(mF2(xS) / rF2 - 1) + abs(pF3(xS) / rF3 - 1) + abs(pF4(xS) / rF4 - 1) + abs(
71     mF5(xS) / rF5 - 1) + abs(mF6(xS) / rF6 - 1)) / 5 * 100;
72
73 formatter = '— FS —\n%s\n%s\n%s\n%s\n%s\n\n';
74 s1 = sprintf('FS = %.3f', resultS);
75 s2 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', xS, sum(xS));
76 s3 = sprintf('F2 = %.3f, F3 = %.3f, F4 = %.3f, F5 = %.3f, F6 = %.3f', -mF2(xS), pF3(xS),
77     pF4(xS), -mF5(xS), -mF6(xS));
78 s4 = sprintf('F2/rF2 = %.3f%%, F3/rF3 = %.3f%%, F4/rF4 = %.3f%%, F5/rF5 = %.3f%%, F6/rF6
79     = %.3f%%', mF2(xS) / rF2 * 100, pF3(xS) / rF3 * 100, pF4(xS) / rF4 * 100, mF5(xS) /
80     rF5 * 100, mF6(xS) / rF6 * 100);
81 s5 = sprintf('Mean = %.3f%%', mean);
82 fprintf(formatter, s1, s2, s3, s4, s5);
83
84 for i = 1 : 1 : sizeK
85     mean = (abs(mF2(xV(:, i)) / rF2 - 1) + abs(pF3(xV(:, i)) / rF3 - 1) + abs(pF4(xV(:, i)
86     )) / rF4 - 1) + abs(mF5(xV(:, i)) / rF5 - 1) + abs(mF6(xV(:, i)) / rF6 - 1)) / 5 *
87     100;
88
89     formatter = '— %s —\n%s\n%s\n%s\n%s\n\n';
90     s1 = sprintf('a = 0.%d, Ka = %.3f', i, Ka(i));
91     s2 = sprintf('x1 = %.3f, x2 = %.3f, x3 = %.3f, x1 + x2 + x3 = %.3f', xV(:, i), sum(xV
92     (:, i)));
93     s3 = sprintf('F2 = %.3f, F3 = %.3f, F4 = %.3f, F5 = %.3f, F6 = %.3f', -mF2(xV(:, i)),
94     pF3(xV(:, i)), pF4(xV(:, i)), -mF5(xV(:, i)), -mF6(xV(:, i)));
95     s4 = sprintf('F2/rF2 = %.3f%%, F3/rF3 = %.3f%%, F4/rF4 = %.3f%%, F5/rF5 = %.3f%%, F6/
96     rF6 = %.3f%%', mF2(xV(:, i)) / rF2 * 100, pF3(xV(:, i)) / rF3 * 100, pF4(xV(:, i)) /
97     rF4 * 100, mF5(xV(:, i)) / rF5 * 100, mF6(xV(:, i)) / rF6 * 100);
98     s5 = sprintf('Mean = %.3f%%', mean);
99     fprintf(formatter, s1, s2, s3, s4, s5);
100 end

```

```

1 function [c, ceq] = m8f(X)
2     global K;
3
4     M1 = 1;
5     M2 = 1;
6     M3 = 1;
7     D1 = M1 / 2;
8     D2 = M2 / 2;
9     D3 = M3 / 2;
10
11     c = M1 * X(1) + M2 * X(2) + M3 * X(3) + K * sqrt(D1 * (X(1) ^ 2) + D2 * (X(2) ^ 2) +
12     D3 * (X(3) ^ 2)) - 80;
13     ceq = [];
14 end

```

Результат решения задачи:

```

1  — FS —
2  FS = 0.243
3  x1 = 35.000, x2 = 24.145, x3 = 1.000, x1 + x2 + x3 = 60.145
4  F2 = 5589.230, F3 = 3684.450, F4 = 6.650, F5 = 188.602, F6 = 13241.479
5  F2/rF2 = 76.998%, F3/rF3 = 138.513%, F4/rF4 = 100.553%, F5/rF5 = 95.021%, F6/rF6 =
   80.242%
6  Mean = 17.361%
7
8  — a = 0.1, Ka = -1.282 —
9  x1 = 35.000, x2 = 24.145, x3 = 1.000, x1 + x2 + x3 = 60.145
10 F2 = 5589.229, F3 = 3684.451, F4 = 6.650, F5 = 188.601, F6 = 13241.473
11 F2/rF2 = 76.998%, F3/rF3 = 138.513%, F4/rF4 = 100.553%, F5/rF5 = 95.020%, F6/rF6 =
   80.242%
12 Mean = 17.361%
13
14 — a = 0.2, Ka = -0.842 —
15 x1 = 35.000, x2 = 24.144, x3 = 1.000, x1 + x2 + x3 = 60.145
16 F2 = 5589.229, F3 = 3684.450, F4 = 6.650, F5 = 188.602, F6 = 13241.477
17 F2/rF2 = 76.998%, F3/rF3 = 138.513%, F4/rF4 = 100.553%, F5/rF5 = 95.021%, F6/rF6 =
   80.242%
18 Mean = 17.361%
19
20 — a = 0.3, Ka = -0.524 —
21 x1 = 35.000, x2 = 24.144, x3 = 1.000, x1 + x2 + x3 = 60.145
22 F2 = 5589.229, F3 = 3684.450, F4 = 6.650, F5 = 188.602, F6 = 13241.477
23 F2/rF2 = 76.998%, F3/rF3 = 138.513%, F4/rF4 = 100.553%, F5/rF5 = 95.021%, F6/rF6 =
   80.242%
24 Mean = 17.361%
25
26 — a = 0.4, Ka = -0.253 —
27 x1 = 35.000, x2 = 24.144, x3 = 1.000, x1 + x2 + x3 = 60.145
28 F2 = 5589.229, F3 = 3684.450, F4 = 6.650, F5 = 188.602, F6 = 13241.477
29 F2/rF2 = 76.998%, F3/rF3 = 138.513%, F4/rF4 = 100.553%, F5/rF5 = 95.021%, F6/rF6 =
   80.242%
30 Mean = 17.361%
31
32 — a = 0.5, Ka = 0.000 —
33 x1 = 35.000, x2 = 24.145, x3 = 1.000, x1 + x2 + x3 = 60.145
34 F2 = 5589.230, F3 = 3684.450, F4 = 6.650, F5 = 188.602, F6 = 13241.479
35 F2/rF2 = 76.998%, F3/rF3 = 138.513%, F4/rF4 = 100.553%, F5/rF5 = 95.021%, F6/rF6 =
   80.242%
36 Mean = 17.361%
37
38 — a = 0.6, Ka = 0.253 —
39 x1 = 35.000, x2 = 24.145, x3 = 1.000, x1 + x2 + x3 = 60.145
40 F2 = 5589.230, F3 = 3684.450, F4 = 6.650, F5 = 188.602, F6 = 13241.479
41 F2/rF2 = 76.998%, F3/rF3 = 138.513%, F4/rF4 = 100.553%, F5/rF5 = 95.021%, F6/rF6 =
   80.242%
42 Mean = 17.361%
43
44 — a = 0.7, Ka = 0.524 —
45 x1 = 35.000, x2 = 24.145, x3 = 1.000, x1 + x2 + x3 = 60.145
46 F2 = 5589.230, F3 = 3684.450, F4 = 6.650, F5 = 188.602, F6 = 13241.479
47 F2/rF2 = 76.998%, F3/rF3 = 138.513%, F4/rF4 = 100.553%, F5/rF5 = 95.021%, F6/rF6 =
   80.242%
48 Mean = 17.361%
49
50 — a = 0.8, Ka = 0.842 —
51 x1 = 35.000, x2 = 20.002, x3 = 1.000, x1 + x2 + x3 = 56.002
52 F2 = 4523.611, F3 = 3270.208, F4 = 6.633, F5 = 180.361, F6 = 11347.389
53 F2/rF2 = 62.318%, F3/rF3 = 122.940%, F4/rF4 = 100.308%, F5/rF5 = 90.869%, F6/rF6 =
   68.764%
54 Mean = 20.259%
55
56 — a = 0.9, Ka = 1.282 —

```

⁵⁷ $x_1 = 34.155$, $x_2 = 14.355$, $x_3 = 0.000$, $x_1 + x_2 + x_3 = 48.510$
⁵⁸ $F_2 = 1781.719$, $F_3 = 2460.145$, $F_4 = 6.588$, $F_5 = 131.020$, $F_6 = 5670.774$
⁵⁹ $F_2/rF_2 = 24.545\%$, $F_3/rF_3 = 92.487\%$, $F_4/rF_4 = 99.619\%$, $F_5/rF_5 = 66.010\%$, $F_6/rF_6 = 34.364\%$
⁶⁰ $\text{Mean} = 36.595\%$

Результаты решения для критерия в виде таблицы:

α	K_α	x_1, Mill	x_2, Mill	x_3, Mill	$F_2, \text{Mill\$}$	$F_3, \text{Mill\$}$	$F_4, \text{Mill\$}$	$F_5, \text{Mill\$}$	$F_6, \text{Mill\$}$	Средняя разница %
det	det	35	24.145	1	5589.230	3684.450	6.650	188.602	13241.479	17.361
0.1	-1.282	35	24.145	1	5589.230	3684.450	6.650	188.602	13241.479	17.361
0.2	-0.842	35	24.145	1	5589.230	3684.450	6.650	188.602	13241.479	17.361
0.3	-0.524	35	24.145	1	5589.230	3684.450	6.650	188.602	13241.479	17.361
0.4	-0.253	35	24.145	1	5589.230	3684.450	6.650	188.602	13241.479	17.361
0.5	0	35	24.145	1	5589.230	3684.450	6.650	188.602	13241.479	17.361
0.6	0.253	35	24.145	1	5589.230	3684.450	6.650	188.602	13241.479	17.361
0.7	0.524	35	24.145	1	5589.230	3684.450	6.650	188.602	13241.479	17.361
0.8	0.842	35	20.002	1	4523.611	3270.208	6.633	180.361	11347.389	20.259
0.9	1.282	34.155	14.355	0	1781.719	2460.145	6.588	131.020	5670.774	36.595

* Функция F_1 используется только для вычисления критерия F_6 , поэтому не рассчитывается отдельно.

Увеличение доверительной вероятности α приводит к ухудшению результатов решения. Неизменность результатов на промежутке $\alpha = [0.1, 0.7]$ объясняется выбором ограничения.

1.5 Вывод

Можно заметить, что аддитивная и мультипликативная свертка выдают одинаковый, наиболее оптимальный результат. Наилучший результат этих методов обусловлен наличием коэффициентов значимости.

Методы, не подразумевающие введение весовых коэффициентов показывают похожий результат, который в целом хуже, чем у аддитивной и мультипликативной свертки.