Федеральное государственное образовательное бюджетное учреждение
высшего образования
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»

Департамент анализа данных, принятия решений
и финансовых технологий

Е.Ф. Олехова

МЕТОДЫ ОПТИМАЛЬНЫХ РЕШЕНИЙ

Сборник заданий для самостоятельной работы с использованием R

Для бакалавров направления 39.03.01 «Социология»
Профиль «Экономическая социология»

Москва 2019

Федеральное государственное образовательное бюджетное учреждение высшего образования

«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»

Департамент анализа данных, принятия решений и финансовых технологий

Е.Ф. Олехова

МЕТОДЫ ОПТИМАЛЬНЫХ РЕШЕНИЙ

Сборник заданий для самостоятельной работы с использованием R

Одобрен на заседании Совета департамента анализа данных, принятия решений и финансовых технологий (протокол №10 от 19.03.2019 г.)

Электронное издание

Москва 2019

УДК 519.816(076.1)
ББК 22.18
О – 53

Рецензент: **И.В. Орлова** – к.э.н., профессор Департамента анализа данных, принятия решений и финансовых технологий

Олехова Е.Ф. Методы оптимальных решений. Сборник заданий с использованием R. – М.: Финансовый университет при Правительстве РФ, 2019. - 44с.

Сборник заданий содержит ряд задач, традиционно изучаемых в рамках курсов принятия оптимальных решений, и фрагменты программ их исследования и решения на языке R. Сборник заданий предназначен для организации самостоятельной работы студентов на семинарах и для домашних контрольных работ.

Сборник заданий может быть полезен студентам всех специальностей, стремящимся познакомиться с современными вычислительными технологиями, а также аспирантам, магистрантам, преподавателям.

УДК 519.816(076.1)
ББК 22.18

Учебное издание
Олехова Елена Федоровна
Методы оптимальных решений. Сборник заданий с импользованием R.

Компьютерный набор, верстка: Е.Ф. Олехова
Формат 60x90/16. Гарнитура Times New Roman
Усл. п.л. 2,75. Изд. № – 2019. Тираж – 0 экз.
Заказ №
Отпечатано в Финансовом университете

Содержание

Предисловие 5
Задание 1. Модель Леонтьева 6
Задание 2. Задача оптимизации прибыли фирмы-монополиста 11
Задание 3. Задача оптимального выбора потребителя 15
Задание 4. Задача о максимизации прибыли с бюджетным ограничением 19
Задание 5. Задача линейного программирования. Графический метод 23
Задание 6. Задача об оптимальном использовании ресурсов 29
Задание 7. Транспортная задача 34
Задание 8. Задача теории игр 40
Литература 44

Предисловие

Сборник заданий предназначен для студентов направления «Социология», профиль «Экономическая социология», изучающих дисциплину «Методы оптимальных решений» как дисциплину по выбору.

В сборник заданий включены задачи, традиционно изучаемые в рамках дисциплины «Методы оптимальных решений» и других дисциплин, связанных с моделями принятия оптимальных решений: задачи линейного и нелинейного программирования и сводящиеся к ним. В том числе задача оптимального выбора потребителя, задача о максимизации прибыли при наличии ограничений, задача об оптимальном использовании ресурсов транспортная задача, задача теории игр и др. Для исследования и решения этих задач приведены фрагменты программ на языке R. Предлагаемые задания могут быть использованы для работы в компьютерном классе или в качестве заданий Домашней контрольной работы.

Сборник заданий разработан с целью развития у студентов способности применять современные математические методы для решения стандартных теоретических и прикладных задач, интерпретировать полученные результаты, оформлять аналитические и отчетные материалы по результатам работы. При этом студенты знакомятся с компьютерными технологиями реализации математических методов и моделей для описания и исследования прикладных задач, учатся использовать компьютерные методы представления данных и средства визуализации количественных данных в R. Студенты также приобретают навыки вычислительной работы в RStudio и графической визуализации результатов исследований.

Сборник заданий может быть полезен студентам всех специальностей, стремящимся познакомиться с современными вычислительными технологиями и использовать язык R в своих курсовых, дипломных и научных работах.

Задание 1. Модель Леонтьева [1, 2 – гл. 2]

Дана балансовая таблица в двухотраслевой модели Леонтьева. Постройте структурную матрицу и проверьте её продуктивность.

Вычислите необходимый объем валового выпуска каждой отрасли, если конечное потребление первой отрасли увеличится вдвое, а второй сохранится на прежнем уровне. Найдите чистую продукцию отраслей.

Проверить результаты вычислений на компьютере.

Данные к заданию 1

Вариант		Вариант	
1.	$\begin{array}{ccccc} & I & II & y & x \\ I & 11 & 25 & 90 & 126 \\ II & 15 & 20 & 130 & 165 \end{array}$	2.	$\begin{array}{ccccc} & I & II & y & x \\ I & 12 & 26 & 95 & 133 \\ II & 18 & 24 & 131 & 173 \end{array}$
3.	$\begin{array}{ccccc} & I & II & y & x \\ I & 13 & 38 & 100 & 151 \\ II & 24 & 30 & 160 & 214 \end{array}$	4.	$\begin{array}{ccccc} & I & II & y & x \\ I & 14 & 40 & 120 & 174 \\ II & 20 & 28 & 150 & 198 \end{array}$
5.	$\begin{array}{ccccc} & I & II & y & x \\ I & 5 & 15 & 60 & 80 \\ II & 14 & 16 & 90 & 120 \end{array}$	6.	$\begin{array}{ccccc} & I & II & y & x \\ I & 6 & 25 & 90 & 121 \\ II & 15 & 20 & 130 & 165 \end{array}$
7.	$\begin{array}{ccccc} & I & II & y & x \\ I & 7 & 26 & 95 & 128 \\ II & 18 & 24 & 131 & 173 \end{array}$	8.	$\begin{array}{ccccc} & I & II & y & x \\ I & 8 & 38 & 100 & 146 \\ II & 24 & 30 & 160 & 214 \end{array}$
9.	$\begin{array}{ccccc} & I & II & y & x \\ I & 9 & 40 & 120 & 169 \\ II & 20 & 28 & 150 & 198 \end{array}$	10.	$\begin{array}{ccccc} & I & II & y & x \\ I & 10 & 15 & 60 & 85 \\ II & 14 & 16 & 90 & 120 \end{array}$
11.	$\begin{array}{ccccc} & I & II & y & x \\ I & 11 & 30 & 90 & 131 \\ II & 15 & 20 & 130 & 165 \end{array}$	12.	$\begin{array}{ccccc} & I & II & y & x \\ I & 12 & 31 & 95 & 138 \\ II & 18 & 24 & 131 & 173 \end{array}$
13.	$\begin{array}{ccccc} & I & II & y & x \\ I & 13 & 35 & 100 & 148 \\ II & 24 & 30 & 160 & 214 \end{array}$	14.	$\begin{array}{ccccc} & I & II & y & x \\ I & 14 & 42 & 120 & 176 \\ II & 20 & 28 & 150 & 198 \end{array}$
15.	$\begin{array}{ccccc} & I & II & y & x \\ I & 5 & 16 & 60 & 81 \\ II & 4 & 16 & 90 & 120 \end{array}$	16.	$\begin{array}{ccccc} & I & II & y & x \\ I & 6 & 25 & 90 & 121 \\ II & 16 & 20 & 130 & 166 \end{array}$
17.	$\begin{array}{ccccc} & I & II & y & x \\ I & 7 & 22 & 95 & 124 \\ II & 17 & 24 & 131 & 172 \end{array}$	18.	$\begin{array}{ccccc} & I & II & y & x \\ I & 8 & 25 & 100 & 133 \\ II & 18 & 30 & 160 & 208 \end{array}$
19.	$\begin{array}{ccccc} & I & II & y & x \\ I & 9 & 27 & 120 & 156 \\ II & 19 & 28 & 150 & 197 \end{array}$	20.	$\begin{array}{ccccc} & I & II & y & x \\ I & 10 & 29 & 60 & 99 \\ II & 20 & 28 & 90 & 138 \end{array}$
21.	$\begin{array}{ccccc} & I & II & y & x \\ I & 11 & 25 & 90 & 126 \\ II & 21 & 23 & 130 & 174 \end{array}$	22.	$\begin{array}{ccccc} & I & II & y & x \\ I & 12 & 26 & 95 & 133 \\ II & 22 & 28 & 131 & 181 \end{array}$
23.	$\begin{array}{ccccc} & I & II & y & x \\ I & 13 & 38 & 100 & 151 \\ II & 23 & 36 & 160 & 219 \end{array}$	24.	$\begin{array}{ccccc} & I & II & y & x \\ I & 14 & 40 & 120 & 174 \\ II & 24 & 30 & 150 & 204 \end{array}$

Вариант		I	II	y	x
25.	I	12	15	100	127
	II	25	16	150	191
26.	I	13	25	90	128
	II	26	20	130	176
27.	I	7	26	95	128
	II	17	24	131	172
28.	I	10	38	100	148
	II	18	30	160	208
29.	I	9	35	120	164
	II	15	28	150	193
30.	I	10	28	60	98
	II	16	19	90	125
31.	I	11	30	100	141
	II	17	20	130	167

Фрагменты программы и результатов исследования модели Леонтьева

```
# Модель Леонтьева. Вариант 31
X <- rbind(c(11,30),c(17,20))  # Составление матрицы из двух строк-векторов
y <- cbind(c(100,130))      # Вектор-столбец конечного потребления
x <- cbind(c(141,167))      # Вектор-столбец объемов производства (валового выпуска)
y2 <- cbind(c(200,130))     # Новый вектор-столбец конечного потребления
X;y;x;y2                    # Вывод матрицы X, векторов y,x,y2
```

```
     [,1] [,2]
[1,]   11   30
[2,]   17   20
     [,1]
[1,]  100
[2,]  130
     [,1]
[1,]  141
[2,]  167
     [,1]
[1,]  200
[2,]  130
```

```
E <- diag(2);E              # Единичная матрица 2x2
```

```
     [,1] [,2]
[1,]    1    0
[2,]    0    1
```

```
A <- rbind(c(0,0),c(0,0))  # Составление матрицы из двух строк-векторов
A[1,1] <- X[1,1]/x[1] # Построение матрицы прямых затрат A
A[2,1] <- X[2,1]/x[1]
A[1,2] <- X[1,2]/x[2]
A[2,2] <- X[2,2]/x[2]
A                    # Вывод матрицы прямых затрат A
```

```
           [,1]      [,2]
[1,] 0.07801418 0.1796407
[2,] 0.12056738 0.1197605
```

```
EA <- E-A; EA        # Построение матрицы  (E-A) и ее вывод
```

```
           [,1]       [,2]
[1,]  0.9219858 -0.1796407
[2,] -0.1205674  0.8802395
```

```
S <- solve(EA); S   # Построение матрицы полных затрат S=(E-A)^-1
```

```
          [,1]      [,2]
[1,] 1.1143548 0.2274194
[2,] 0.1526344 1.1672043
```

```
E1 <- S%*%EA;E1 # Проверка обратной матрицы
```

```
     [,1]         [,2]
[1,]    1 2.775558e-17
[2,]    0 1.000000e+00
```

```
S>0       # Проверка каждого элемента матрицы S на положительность
```

```
     [,1] [,2]
[1,] TRUE TRUE
[2,] TRUE TRUE
```

```
all(S>0) # Ответ на вопрос: для всех ли элементов матрицы S выполнено
 ↪  условие
```

```
[1] TRUE
```

```r
if(all(S>0) ) print ("Матрица А продуктивна")
```

```
[1] "Матрица А продуктивна"
```

```r
eigen(A)# Вычисление собственных значений и собств. векторов матрицы А
eigen() decomposition
```

```
$values
[1]  0.24752951 -0.04975485

$vectors
           [,1]       [,2]
[1,] -0.7273068 -0.8149028
[2,] -0.6863124  0.5795977
```

```r
d <- eigen(A)\$values; d   # Собственные значения матрицы А
```

```
[1]  0.24752951 -0.04975485
```

```r
lambda_A<- max(d);lambda_A  # Определение числа Фробениуса
```

```
[1] 0.2475295
```

```r
P <- eigen(A)$vectors; P # Собств. векторы А, стоящие в столбцах матр. Р
```

```
           [,1]       [,2]
[1,] -0.7273068 -0.8149028
[2,] -0.6863124  0.5795977
```

```r
x1 <- S %*% y; x1   # Вектор объемов производства по отраслям (x1=x)
```

```
     [,1]
[1,]  141
[2,]  167
```

```r
x2 <- S %*% y2; x2  # Вектор новых объемов производства по отраслям (валового выпуска)
```

```
        [,1]
[1,] 252.4355
[2,] 182.2634
```

```
# Чистая продукция отрасли - разность между валовой продукцией этой
# отрасли и продукцией всех отраслей на производство этой отрасли
xc <- cbind(c(0,0))     # Вектор-столбец  выпуска чистой продукции
x11<-A[1,1]*x2[1]
x21<-A[2,1]*x2[1]
xc[1]<-x2[1]-(x11+x21)
x12<-A[1,2]*x2[2]
x22<-A[2,2]*x2[2]
xc[2]<-x2[2]-(x12+x22); xc
```

```
        [,1]
[1,] 202.3065
[2,] 127.6935
```

Ответ. Матрица прямых затрат $A = \begin{pmatrix} 0.07801418 & 0.1796407 \\ 0.12056738 & 0.1197605 \end{pmatrix}$,
матрица A продуктивна,
число Фробениуса $\lambda_A = 0.2475295$,
вектор Фробениуса $p_A = (0.7273068, 0.6863124)$,
новый вектор объемов производства (валового выпуска) $x2 = (252.4355; 182.2634)$,
вектор чистой продукции отраслей $xc = (202.3065; 127.6935)$.

Задание 2. Оптимизация прибыли фирмы-монополиста [1,2 – гл.9]

Для товаров x_1, x_2 известны функции спроса $q_1 = a + bp_1$ и $q_2 = c + dp_2$, где p_1, p_2 – цены единицы товаров x_1, x_2 соответственно.

Фирма монополист имеет функцию издержек $C = kq_1^2 + lq_1q_2 + mq_2^2 + n$.

1. Вычислите максимальную прибыль фирмы $\Pi(q_1, q_2) = p_1q_1 + p_2q_2 - C(q_1, q_2)$ в этих условиях и найдите соответствующий оптимальный производственный план.

2. Проверьте результаты вычислений на компьютере.

Данные к заданию 2

Вариант	a	b	c	d	k	l	m	n
1.	90	-1	62	-2	7	5	1	4
2.	42	-1	30	-1	5	3	1	7
3.	68	-1	96	-2	4	2	4	3
4.	53	-1	78	-4	4	3	2	8
5.	58	-2	28	-1	3	5	2	9
6.	82	-2	39	-1	3	5	2	8
7.	76	-4	54	-1	2	2	4	9
8.	60	-2	56	-2	3	4	2	8
9.	70	-2	47	-1	2	5	3	4
10.	62	-2	66	-2	3	5	4	11
11.	27	-1	68	-2	1	3	2	10
12.	70	-1	42	-1	4	5	2	6
13.	65	-1	68	-2	4	5	1	4
14.	85	-1	68	-2	6	5	1	4
15.	28	-1	20	-1	4	2	1	5
16.	66	-1	42	-2	4	2	1	3
17.	80	-2	52	-1	2	6	3	7
18.	70	-2	47	-1	2	5	3	5
19.	48	-1	116	-4	3	4	2	7
20.	66	-1	96	-4	5	3	2	8
21.	68	-2	34	-1	3	5	2	3
22.	82	-2	39	-1	3	5	2	8
23.	96	-4	40	-1	2	5	4	9
24.	102	-2	80	-2	3	4	2	5
25.	60	-2	42	-1	2	5	3	4
26.	74	-2	50	-2	4	5	2	9
27.	63	-1	90	-2	3	5	2	4
28.	92	-2	80	-4	6	5	1	9
29.	94	-4	23	-1	4	3	1	8
30.	56	-1	74	-2	4	2	4	9
31.	80	-2	58	-1	2	5	3	7

Фрагменты программы и результатов исследования задачи о максимизации прибыли монополиста

```r
# Задача о максимизации прибыли монополиста 31 вар
F <-function(x) {   # Задание функции для минимизации
   q1 <-x[1]
   q2 <-x[2]
   -(-2.5*q1^2-4*q2^2-5*q1*q2+40*q1+58*q2-7) }
# Функция, вычисляющая частные производные (градиент) функции:
gr <- function(x) {
   q1 <- x[1]
   q2 <- x[2]
  c(-(40-5*q1-5*q2),
     -(58-5*q1-8*q2)) }
# Безусловная минимизация
res <- optim(c(0.5,0.5), F, gr, method = "BFGS")
res      # Вывод результатов минимизации
```

```
$par
[1] 2.000000 6.000002
$value
[1] -207
$counts
function gradient
      17        5
$convergence
[1] 0
```

```r
# Вычисление градиента и гессиана исследуемой функции в точке экстремума
Deriv2Dn  <- function (expr,x=0, y=0)
{ M <- c(x=x,y=y)
  value <- eval(expr)
  grad <- c(x=0,y=0)
  grad["x"] <- eval(D(expr, "x"))
  grad["y"] <- eval(D(expr, "y"))
  hess <- array(0, c(2L, 2L), list(c("x", "y"), c("x", "y")))
  hess["x","x"] <- eval(D(D(expr, "x"), "x"))
  hess["y","y"] <- eval(D(D(expr, "y"), "y"))
  hess["x","y"] <- hess["y","x"] <- eval(D(D(expr, "x"), "y"))
  Delta1<-hess["x","x"]
  Delta2<- hess["x","x"]*hess["y","y"]-hess["x","y"]^2
  Rez <- list(Function = expr, Point = M, Value_Function= value,
              Gradient= grad, Hessian = hess,
              Delta1=Delta1,Delta2=Delta2)
  return(Rez)}
# Вызов функции Deriv2Dn:
Deriv2Dn(substitute(-2.5*x^2-5*x*y-4*y^2+40*x+58*y-7),x=2,y=6)
```

```
$Function
-2.5 * x^2 - 5 * x * y - 4 * y^2 + 40 * x + 58 * y - 7
$Point
x y
2 6
$Value_Function
[1] 207
$Gradient
x y
0 0
$Hessian
   x  y
x -5 -5
y -5 -8
$Delta1
[1] -5
$Delta2
[1] 15
```

Ответ. Оптимальный план $(q_1^*; q_2^*) = (2; 6)$, максимальная прибыль $\Pi(q_1^*; q_2^*) = 207$.

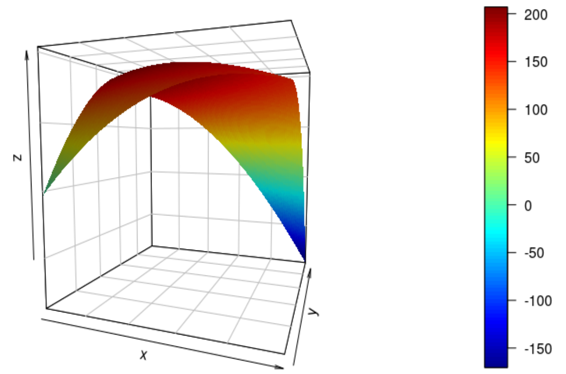

Рис. 1. Функия прибыли монополиста $\Pi(q_1, q_2)$

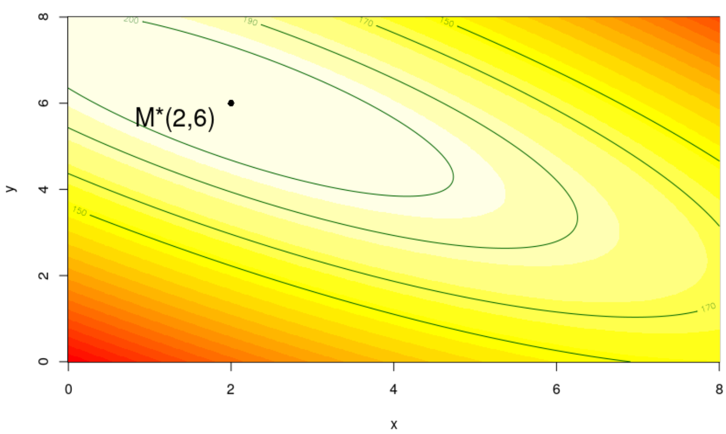

Рис. 2. Линии уровня функции прибыли и точка максимума

```
#Геометрическая интерпретация модели
#install.packages("plot3D",dependencies=TRUE)
require(plot3D)   # То же, что и library(plot3D)
Loading required package: plot3D
# Функция прибыли
M <- mesh(seq(0, 10, length.out = 100), seq(0,10,
     length.out = 100)) # Создаем сеть (ui,vj)
u <- M$x       # Объявляем значения параметра u
v <- M$y       # Объявляем значения параметра v
x <- u        # Вводим функцию для координаты x поверхности
y <- v        # Вводим функцию для координаты y поверхности
z <- -2.5*x^2-5*x*y-4*y^2+40*x+58*y-7  # Вводим функцию для координаты z
# Функция прибыли - поверхность
surf3D(x, y, z, colvar = z, phi = 5, bty = "b2",theta = 20,
       lighting = TRUE, ltheta = 80, colkey = TRUE, box = TRUE)
# Линии уровня
x  <- seq(0, 8, by=0.01)   # Задаем последовательность значений x
y  <- seq(0, 8, by=0.01)  # Задаем последовательность значений y
f <- function(x,y) {-2.5*x^2-5*x*y-4*y^2+40*x+58*y-7}  # Задаем функцию
 ↪ f
z <- outer(x, y, f) # Вычисляем f во всех точках сетки (x,y)
image(x, y, z, col=  heat.colors(20))
# Линии уровня - разные (подобрать, их значения указаны)
contour(x,y,z, col="darkgreen",add=TRUE,
        levels=c(150,170,190,200),
        method = "edge", vfont = c("sans serif", "plain"))
# Точка максимума прибыли
points(2,6,pch=16,cex = 1, col = "black")
# Надпись к точке, adj=c(..., ...) - отклонение от координат точки
text(2,6, "M*(2,6)", cex = 1.75,
     adj = c(1.2,1.2))
```

Задание 3. Задача оптимального выбора потребителя [1, 2 – гл.8]

Функция полезности потребителя для двух товаров имеет вид $U = U(x, y)$, где x, y – количества приобретаемых товаров.

1. Определите максимальную полезность и оптимальный набор товаров, если доход потребителя – I ден.ед., а цены товаров – p, q ден.ед. соответственно.
2. Постройте график функции полезности.
3. Изобразите допустимое множество, кривые безразличия и оптимальную точку.
4. Найдите уравнение кривой безразличия, на которой находится потребитель в оптимальной точке.
5. Определите норму замены второго товара первым в оптимальной точке.
6. Найдите функцию спроса для первого товара и постройте ее график.
7. Определите эластичность спроса на первый товар по цене.
8. Проверьте результаты вычислений на компьютере.

Данные к заданию 3

Вариант	$U(x,y)$	p	q	I
1.	$5x^{\frac{2}{3}}y^{\frac{1}{2}}$	16	17	2396
2.	$5(x-7)y^{\frac{5}{6}}$	7	2	1694
3.	$2x^{\frac{4}{7}}y$	5	11	1055
4.	$8xy^{\frac{3}{7}}$	18	7	3368
5.	$7(xy)^{\frac{2}{5}}$	3	16	1847
6.	$6(x+5)^{\frac{7}{8}}(y-6)^{\frac{2}{3}}$	19	18	1556
7.	$9(x-4)^{\frac{1}{2}}(y+8)^{\frac{1}{2}}$	11	2	2908
8.	$7(x-7)^{\frac{1}{3}}y^{\frac{2}{3}}$	5	9	1845
9.	$7(x+5)^{\frac{3}{4}}y^{\frac{2}{3}}$	11	2	1139
10.	$5x^{\frac{2}{5}}y$	2	7	2619
11.	$4x^{\frac{5}{9}}(y-9)^{\frac{2}{3}}$	8	11	2519
12.	$5(x-4)^{\frac{2}{3}}y^{\frac{5}{8}}$	4	11	3037
13.	$7\ln(x-6)+3\ln(y-2)$	17	2	2108
14.	$4(x-3)^{\frac{5}{6}}(y+2)^{\frac{1}{6}}$	15	4	2943
15.	$3x(y-7)^{\frac{3}{4}}$	2	19	2070
16.	$3x^{\frac{5}{7}}(y-7)^{\frac{5}{7}}$	5	8	2336
17.	$7(x-1)^{\frac{3}{8}}y^{\frac{5}{8}}$	9	2	3056
18.	$x^{\frac{2}{5}}y^{\frac{2}{5}}$	7	5	1684
19.	$5(x+9)^{\frac{1}{2}}(y+5)^{\frac{1}{2}}+4$	19	13	2574
20.	$2x^{\frac{8}{9}}y^{\frac{8}{9}}$	2	5	2349
21.	$9\ln(x+4)+4\ln(y+9)$	4	11	1873
22.	$8\ln x+3\ln(y-3)$	9	2	991
23.	$\ln(x-7)+8\ln(y-1)$	11	16	2444
24.	$6\ln x+7\ln(y+6)$	19	9	3174
25.	$5\ln(x+6)+2\ln(y-8)$	11	14	2019
26.	$7\ln(x+8)+2\ln(y-7)$	16	9	1844
27.	$8\ln x+5\ln(y-4)$	17	2	1626
28.	$7\ln(x-6)+3\ln(y-2)$	17	2	2108
29.	$7\ln(x+8)+2\ln(y-7)$	16	9	1844
30.	$2(xy)^{\frac{3}{8}}$	13	19	1721
31.	$5\ln(x-3)+8\ln(y-3)$	15	4	422

Вариант 31.

Исходные данные к задаче:

$$U(x, y) = 5\ln(x-3) + 8\ln(y-3),\ \ I = 422,\ \ p = 15,\ \ q = 4.$$

Математическая постановка задачи оптимального выбора потребителя:

$$U(x, y) = 5\ln(x-3) + 8\ln(y-3) \to \max_{(x,y)\in D},$$

$$D: 15x + 4y \le 422,\ \ x > 3,\ \ y > 3.$$

Ответ. $M^*(12,36; 59,15)$; $U(M^*) = 43,41$; $x = \frac{24}{13} + \frac{2050}{13p}$; $3,75$; $-0,85$.

Фрагменты программы и результатов исследования задачи об оптимальном выборе потребителя

```r
# Задача оптимального выбора потребителя. Вариант 31
F <-function(x) {
  x1 <-x[1]
  x2 <-x[2]
  -5 * log(x1-3)-8*log(x2-3)
}
#  Функция, вычисляющая частные производные (градиент):
gr <- function(x) {
  x1 <- x[1]
  x2 <- x[2]
  c(-5 /(x1-3),
    -8/(x2-3))
}
# Оптимизация
constrOptim(c(10,50), F, gr,
            ui=rbind(c(-15,-4),c(1,0),c(0,1)), ci=c(-422,3,3))
```

```
$par
[1] 12.39700 59.01124
$value
[1] -43.40637
$counts
function gradient
     109       44

$convergence
[1] 0
$message
NULL

$outer.iterations
[1] 3
$barrier.value
[1] -0.06224257
```

```r
# Геометрическая интерпретация задачи
#install.packages("plot3D",dependencies=TRUE)
require(plot3D)   # То же, что и library(plot3D)
# Функция полезности
M <- mesh(seq(20, 100, length.out = 200),
     seq(10, 40,length.out = 100)) # Создаем сеть (ui,vj)
u <- M$x       # Объявляем значения параметра u
v <- M$y       # Объявляем значения параметра v
x <- u        # Вводим функцию для координаты x поверхности
y <- v        # Вводим функцию для координаты y поверхности
z <- 5*log(x-3)+8*log(y-3)  # Вводим функцию для координаты z
surf3D(x, y, z, colvar = z, phi = 5, bty = "b2",theta = 40,
       lighting = TRUE, ltheta = 60, colkey = TRUE, box = TRUE)
# Линии уровня и допустимое множество
x  <- seq(3, 60, by=1)   # Задаем последовательность значений x
y  <- seq(3, 120, by=1)  # Задаем последовательность значений y
f <- function(x,y) {5*log(x-3)+8*log(y-3) }  # Задаем функцию f
z <- outer(x, y, f) # Вычисляем f во всех точках сетки (x,y)
image(x, y, z, col=  heat.colors(12))
# Допустимое множество - треугольник,
#координаты вершин находим отдельно и подставляем
#(первый список - первые координаты вершин,
#второй список - вторые координаты)
polygon(c(3,27.333,3), c(94.25,3,3), density = NA, angle = 60,
 border = c("darkgreen", "yellow"), col ="green", lwd=3, lty="solid")
# Линии уровня - разные (подобрать, их значения указаны)
contour(x,y,z, col="blue4",add=TRUE, levels=c(35,38,40,42,46,48),
        method = "edge", vfont = c("sans serif", "plain"))
# Линия оптимального уровня (определяется после оптимизации)
contour(x,y,z, levels=c(43.41),col="red", lwd=5,add=TRUE,
        method = "edge", vfont = c("sans serif", "plain"))
# Точка касания (ее координаты находятся после того,
#как пройдет оптимизация)
points(12.4,59.01,pch=15,cex = 1, col = "black")
# Функция спроса
# X=24/13+2050/(13*p)
X <- function(x) {
  p <-x
  24/13+2050/(13*p)}
x <- seq(5, 100, by = 1) # Задаем последовательность p
p <- x
y <- X(p)
plot(x,y,type="l",lty=1, fg="pink4",
     xlim=c(0,100),ylim=c(0,40),ylab="X(p)",xlab="p",
     lwd = 3,col="violetred3",
     col.axis="palevioletred4",
     main = "Функция спроса",col.main = "red")
abline(h = 0, v = 0, col = "gray40") # Рисуем оси координат
```

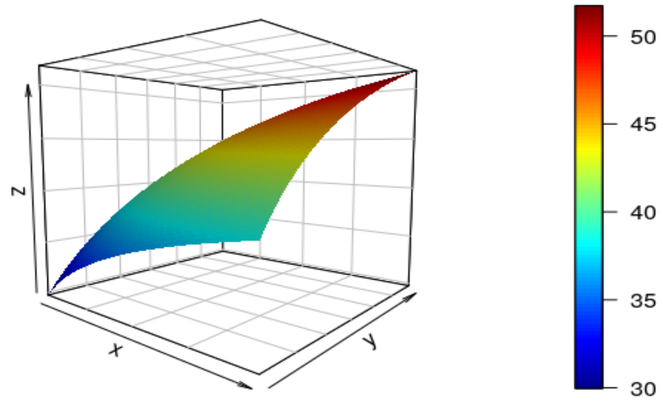

Рис. 3. Функия полезности $U(x, y) = 5\ln(x-3) + 8\ln(y-3)$

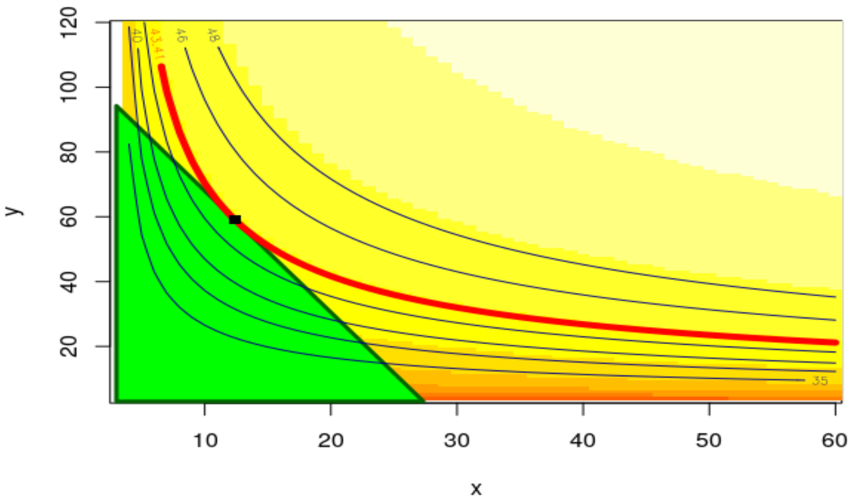

Рис. 4. Линии уровня функции полезности и бюджетное ограничение

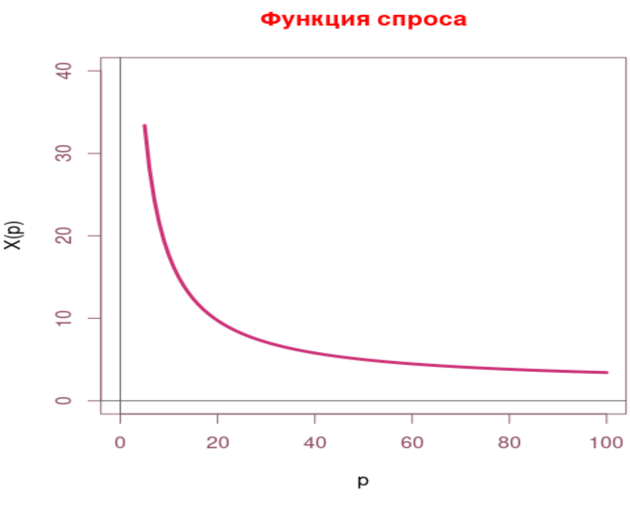

Рис. 5. Функия спроса

Задание 4. Задача максимизации прибыли производителя [1, 2 – гл.9]

Производственная функция (в ден. выражении) имеет вид $Q(x,y) = Ax^{a_1}y^{a_2}$, где x, y – количества единиц первого и второго ресурсов. Стоимость единицы первого ресурса – w_1, второго – w_2 (ден.ед.).

1. Найдите максимальную прибыль и оптимальный план.

2. В силу бюджетных ограничений на ресурсы может быть потрачено не более I (ден.ед.). Найдите максимальную прибыль при наличии бюджетных ограничений и оптимальное для производителя сочетание (x, y) количеств используемых ресурсов.

3. Постройте пространственную модель функции прибыли. Постройте карту изоквант.

4. Найдите уравнение изокванты, на которой достигается максимум прибыли при наличии ограничений на издержки. Найдите уравнение изокосты, которая соответствует ограничениям на издержки.

5. Покажите графически, что оптимальной комбинации ресурсов соответствует точка касания найденных изокванты и изокосты.

Данные к заданию 4

Вариант	A	a_1	a_2	w_1	w_2	I
1.	42	1/2	0,25	5	10	100
2.	35	1/2	0,2	4	2	200
3.	45	1/2	1/3	5	8	500
4.	18	1/2	0,4	2	6	150
5.	40	1/3	0,25	4	2	100
6.	38	1/2	0,25	5	4	180
7.	30	1/2	0,2	4	2	120
8.	25	1/2	1/3	2	5	600
9.	13	1/2	0,4	4	1	180
10.	13	1/2	1/3	1	2	500
11.	36	1/3	0,25	4	2	80
12.	40	1/2	0,25	5	8	100
13.	28	1/2	0,2	2	4	200
14.	28	1/2	1/3	3	6	200
15.	12	1/2	0,4	2	4	500
16.	20	1/2	0,3	3	1	1000
17.	56	1/3	0,25	5	3	140
18.	36	1/2	0,25	5	6	100
19.	36	1/2	0,2	4	2	200
20.	34	1/2	1/3	4	8	200
21.	32	1/3	0,25	2	1	160
22.	50	1/3	0,25	6	3	100
23.	32	1/2	0,25	2	5	400
24.	32	1/2	0,2	4	2	150
25.	24	1/2	1/3	2	8	200
26.	21	1/2	1/3	3	1	1000
27.	42	1/3	0,25	3	2	150
28.	22	1/2	1/3	3	1	1200
29.	48	1/3	0,25	5	3	100
30.	18	1/2	1/3	3	1	500
31.	30	1/2	1/3	5	10	600

Вариант 31.
Исходные данные к задаче:

$$Q(x,y) = 30\sqrt{x}\sqrt[3]{y},\ I = 600,\ p = 5,\ q = 10.$$

Математическая постановка задачи о максимизации прибыли с ограничением:

$$Q(x,y) = 30\sqrt{x}\sqrt[3]{y} \to \max_{(x,y)\in D},$$

$$D : 5x + 10y \leq 600,\ x > 0,\ y > 0.$$

Ответ. $x = 72,\ y = 24.$ ◄

Фрагменты программы и результатов исследования задачи о максимизации прибыли производителя

```
# Задача максимизации прибыли с ограничениями
# Функция прибыли с ПФ Кобба-Дугласа
F <-function(x) {
  x1 <-x[1]
  x2 <-x[2]
  -(30 * x1^(1/2) * x2^(1/3)-5*x1-10*x2)}
#  Функция, вычисляющая частные производные (градиент):
gr <- function(x) {
  x1 <- x[1]
  x2 <- x[2]
  c(-15 * x1^(-1/2) * x2^(1/3)+5,
    -10 *x1^(1/2) * x2^(-2/3)+10)}
# Безусловная оптимизация
res <- optim(c(10,50), F, gr, method = "BFGS")
res  # Вывод результатов безусловной оптимизации
```

```
$par
[1] 81.00099 27.00159
$value
[1] -135
$counts
function gradient
      27       26
$convergence
[1] 0
```

```
# Оптимизация с ограничением
constrOptim(c(1,1), fr, grr,
            ui=rbind(c(-5,-10),c(1,0),c(0,1)), ci=c(-600,0,0))
```

```
$par
[1] 71.99909 24.00045
$value
[1] -134.2736
$counts
function gradient
     101       39
$convergence
[1] 0
$outer.iterations
[1] 4
$barrier.value
[1] -0.08881894
```

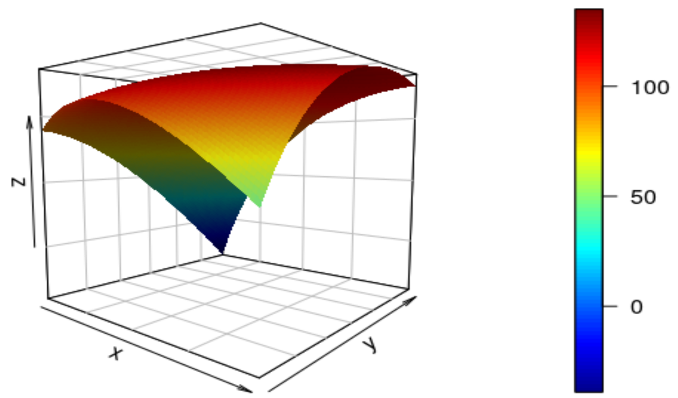

Рис. 6. Производственная функция $Q(x, y) = 30\sqrt{x}\sqrt[3]{y}$

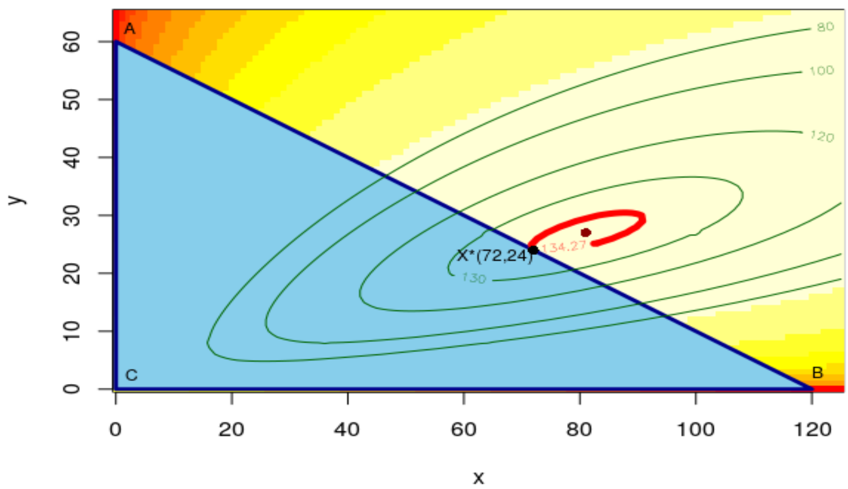

Рис. 7. Изокванты, бюджетное множество и оптимальное решение

```r
# Геометрическая интерпретация задачи
#install.packages("plot3D",dependencies=TRUE)
require(plot3D)   #  library(plot3D)
# Функция прибыли
M <- mesh(seq(20, 100, length.out = 200),
          seq(10, 40, length.out = 100)) # Создаем сеть (ui,vj)
u <- M$x       # Объявляем значения параметра u
v <- M$y       # Объявляем значения параметра v
x <- u        # Вводим функцию для координаты x поверхности
y <- v        # Вводим функцию для координаты y поверхности
z <- 30 * x^(1/2) * y^(1/3)-5*x-10*y  # Вводим функцию для  z
surf3D(x, y, z, colvar = z, phi = 5, bty = "b2",theta = 40,
       lighting = TRUE, ltheta = 60, colkey = TRUE, box = TRUE)
# Линии уровня и допустимое множество
x  <- seq(0, 125, by=1) # Задаем последовательность значений x
y  <- seq(0, 65, by=1)  # Задаем последовательность значений y
f <- function(x,y) { 30 * x^(1/2) * y^(1/3)-5*x-10*y  }
# Задаем функцию f
z <- outer(x, y, f) # Вычисляем f во всех точках сетки (x,y)
image(x, y, z, col=  heat.colors(12))
# Допустимое множество - треугольник, координаты вершин находим
# отдельно и подставляем
#(первый список - первые координаты вершин,
#второй список - вторые координаты)
polygon(c(0,120,0), c(60,0,0), density = NA, angle = 60,
        border = c("darkblue", "yellow"),
        col ="skyblue", lwd=3, lty="solid")
# Линии уровня - разные (подобрать, их значения указаны)
contour(x,y,z, col="darkgreen",add=TRUE,
        levels=c(80,100,120,130,150),
        method = "edge", vfont = c("sans serif", "plain"))
# Линия оптимального уровня (определяется после оптимизации)
contour(x,y,z, levels=c(134.27),col="red", lwd=5,add=TRUE,
        method = "edge", vfont = c("sans serif", "plain"))
# Точка безусловного максимума (без бюджетного ограничения)
# ее координаты находятся после того,
#как пройдет безусловная оптимизация
points(81,27,pch=16,cex = 1, col = "darkred")
# Точка касания, ее координаты находятся после того,
# как пройдет оптимизация
points(72,24,pch=16,cex = 1, col = "black")
# Надписи к точкам, adj=c(..., ...) - отклонение от координат точки
text(72,24, "X*(72,24)", cex = .75,    adj = c(1,1))
text(0,60, "A", cex = .75,     adj = c(-0.7,-0.7))
text(120,0, "B", cex = .75,     adj = c(0,-1))
text(0,0, "C", cex = .75,     adj = c(-0.7,-0.7))
```

Задание 5. Линейное программирование. Графический метод [1,2 – гл. 3, 4]

Решить задачу линейного программирования, используя геометрическую интерпретацию.

Проверить результаты вычислений на компьютере.

Варианты 1-20:

$$F(x_1, x_2) = x_1 + ax_2 \to \max_{x \in D},$$

$$D : \begin{cases} x_1 + 2x_2 \le 10, \\ 3x_1 + 2x_2 \le 18, \\ x_1 - x_2 \ge -b, \\ cx_1 - x_2 \le 8c + 3. \end{cases}$$

Данные к вариантам 1-20:

Номер варианта	Параметры			Номер варианта	Параметры		
	a	b	c		a	b	c
1	5	7	2	11	-5/6	8	1/4
2	1	6	3	12	3	13/2	2
3	-1	6	1/8	13	1	9	1
4	5	9	1	14	-1/3	10	2
5	3/4	7	1	15	7/4	6	3
6	-1/4	10	2	16	-3/4	13/2	1/2
7	4	12	1/2	17	3/2	7	2
8	5/4	9	1/3	18	3	6	1
9	-1	6	1/2	19	4	8	3/4
10	5/6	7	1	20	-1	15/2	1/3

Варианты 21-40:

$$F(x_1, x_2) = ax_1 + x_2 \to \min_{x \in D},$$

$$D : \begin{cases} x_1 + (b-3)x_2 \ge b, \\ (c-4)x_1 + x_2 \ge c, \\ 3x_1 + 2x_2 \ge 11, \\ x_1 \ge 0, x_2 \ge 0 \end{cases}$$

Данные к вариантам 21-40:

Номер варианта	Параметры			Номер варианта	Параметры		
	a	b	c		a	b	c
21	1/4	5	9	31	7/2	5	7
22	5/4	4	6	32	9/2	6	9
23	9/2	7	8	33	1/5	7	7
24	7/4	8	7	34	7/2	4	8
25	5/2	6	6	35	1/3	8	9
26	1/2	7	6	36	1/2	4	9
27	1/6	8	8	37	5/3	8	6
28	5/2	4	7	38	3/4	5	6
29	13/3	5	8	39	1/4	6	8
30	2/3	6	7	40	11/2	7	9

Фрагменты программы и результатов исследования задачи линейного программирования

```
#Задача  ЗЛП,  геометрический метод решения
# Вариант 19
#F(x1, x2) = x1 + 4x2 → max
# A:     x1 + 2x2<= 10,
# B:    3x1 + 2x2<= 18,
# C:     x1-x2>=-8,
# D:  3/4x1-x2<=9.
# Задание системы координат
plot(c(-10,10), c(10,-10), type = "n", xlab = "x", ylab = "y", asp = 1)
# Рисование осей и координатной сетки
abline(h = 0, v = 0, col = "gray60")
abline(h = -10:10, v = -20:20, col = "lightgray", lty = 3)
abline(a = 5, b =-1/2, col = 2) # Прямая-граница  A
text(20,-5, "A", col = 2, adj = c(-.1, -.1))
abline(a = 9, b =-3/2, col = 3) # Прямая-граница B
text(13,-10, "B", col = 3, adj = c(-.1, -.1))
abline(a = 8, b =1, col = 4) # Прямая-граница C
text(-18,-9, "C", col = 4, adj = c(-.1, -.1))
abline(a = -9, b =3/4, col = 5) # Прямая-граница D
text(20,7, "D", col = 5, adj = c(-.1, -.1))
# Область допустимых решений и ее заливка
polygon(c(4,8,-1.2,-18,-2), c(3,-3,-10,-10,6), density = NA, angle = 60,
        #        border = c("darkgreen", "yellow"),
        col ="yellow", lwd=3, lty="solid")
# Граница области - как отрезки (можно рисовать в poligon)
segments(4,3,8,-3,col ="darkgreen", lty = "solid", lwd = 3)
segments(-1.3,-10,8,-3,col ="darkgreen", lty = "solid", lwd = 3)
segments(4,3,-2,6,col ="darkgreen", lty = "solid", lwd = 3)
segments(-18,-10,-2,6,col ="darkgreen", lty = "solid", lwd = 3)
abline(h = -10:10, v = -20:20, col = "lightgray", lty = 3)
# Линии уровня целевой функции
abline(a = 4/4, b =-1/4,lwd=3, col = 2)
text(-20,6, "F=4", col = 2, adj = c(-.1, -.1))
abline(a = 22/4, b =-1/4,lwd=3, col = 2)
text(-14,9, "F=22", col = 2, adj = c(-.1, -.1))
# Нормальный вектор к линии уровня
arrows(-12, 4, -12+1,  4+4,  length = 0.15, angle = 10,
       code = 2, col = 1,lwd=3, lty = par("lty") )
text(-11,6, "n(1,2)", col = 1, adj = c(-.1, -.1))
# Оптимальная точка
points(-2,6,pch=15,cex = 1, col = "black")
text(-2,6, "M*(-2,6)", col = 1, adj = c(-.1, -.1))
```

```r
# Решение ЗЛП симплекс-методом с помощью пакета lpSolveAPI
#install.packages(lpSolveAPI)
library(lpSolveAPI) #обращение к библиотеке
M<-make.lp(ncol=2)  # Объявление количества  переменных в M
name.lp(M,"GEOM-19") # Название модели
colnames(M)<-c("x1","x2") # Названия переменных в модели
lp.control(M,sense="max")$sense # объявление задачи на максимум
set.objfn(M,c(1,4)) # Коэффициенты в целевой функции
add.constraint(M,c(1,2),"<=", 10) #Задается  ограничение A
add.constraint(M,c(3,2),"<=", 18) #Задается  ограничение B
add.constraint(M,c(1,-1),">=", -8) #Задается  ограничение C
add.constraint(M,c(3/4,-1),"<=", 9) #Задается  ограничение D
rownames(M)<-c("A","B","C","D")     # Обозначаются ограничения модели
set.bounds(M,lower=c(-Inf,-Inf),upper=c(Inf,Inf))# Границы для перем.
M   # Построенная модель
```

```
Model name: GEOM-19
            x1    x2
Maximize     1     4
A            1     2  <=  10
B            3     2  <=  18
C            1    -1  >=  -8
D         0.75    -1  <=   9
Kind       Std   Std
Type      Real  Real
Upper      Inf   Inf
Lower     -Inf  -Inf
```

```r
solve.lpExtPtr(M) # Решение задачи
 get.variables(M) # Оптимальный план

```

```
[1] -2  6
```

```r
get.objective(M) # Оптимальное значение целевой функции
```

```
[1] 22
```

```r
x1_opt<-get.variables(M)[1]
x2_opt<-get.variables(M)[2]
F_max<-get.objective(M)
x1_opt;x2_opt      # Оптимальный план
```

```
[1] -2
[1] 6
```

```
66 F_max                # Оптимальное значение целевой функции
```

```
[1] 22
```

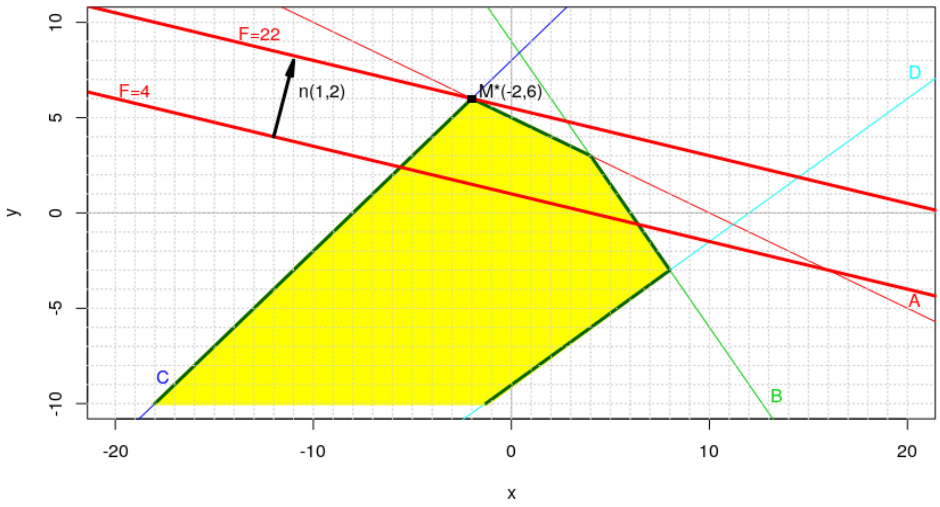

Рис. 8. Геометрический метод решения ЗЛП. Вариант 19.

Ответ. $M^*(-2; 6)$; $F_{max} = F(M^*) = 22$.

```
#Задача ЗЛП, геометрический метод.  Вариант 39
#F(x1, x2) = 1/4x1 + x2 → min
# A:     x1 + 3x2>= 6,
# B:    4x1 + x2>= 8,
# C:    3x1+2x2>=11,
#        x1>=0, x2>=0.
# Задание системы координат
plot(c(-1,10), c(10,-1), type = "n", xlab = "x", ylab = "y", asp = 1)
# Рисование осей и координатной сетки
abline(h = 0, v = 0, col = "gray60")
abline(h = -1:10, v = -1:10, col = "lightgray", lty = 3)
abline(a = 2, b =-1/3, col = 2) # Прямая-граница  A
text(-5,3.8, "A", col = 2, adj = c(-.1, -.1))
abline(a = 8, b =-4, col = 3) # Прямая-граница B
text(-0.8,9, "B", col = 3, adj = c(-.1, -.1))
abline(a = 11/2, b =-3/2, col = 4) # Прямая-граница C
text(-3.1,9, "C", col = 4, adj = c(-.1, -.1))
# Область допустимых решений и ее заливка
polygon(c(0,0,1,3,6,15,15), c(10,8,4,1,0,0,10), density = NA,
      angle = 60,  col ="yellow", lwd=3, lty="solid")
# Граница области - как отрезки (можно рисовать в poligon)
segments(0,10,0,8,col ="darkgreen", lty = "solid", lwd = 3)
segments(0,8,1,4,col ="darkgreen", lty = "solid", lwd = 3)
segments(1,4,3,1,col ="darkgreen", lty = "solid", lwd = 3)
segments(3,1,6,0,col ="darkgreen", lty = "solid", lwd = 3)
segments(15,0,6,0,col ="darkgreen", lty = "solid", lwd = 3)
abline(h = -1:10, v = -1:10, col = "lightgray", lty = 3)
# Линии уровня целевой функции 1/4x1+x2=4 x2=4-1/4x1
abline(a = 4, b =-1/4,lwd=3, col = 2)
text(-2,4.5, "F=4", col = 2, adj = c(-.1, -.1))
# Линии уровня целевой функции 1/4x1+x2=1.5 x2=1.5-1/4x1
abline(a = 1.5, b =-1/4,lwd=3, col = 2)
text(-2,1.2, "F=1.5", col = 2, adj = c(-.1, -.1))
# Нормальный вектор к линии уровня
arrows(-4, 5, -4+1/4,  5+1,  length = 0.15, angle = 10,
       code = 2, col = 1,lwd=3, lty = par("lty") )
text(-6,6, "n(1/4,1)", col = 1, adj = c(-.1, -.1))
points(6,0,pch=15,cex = 1, col = "black")# Оптимальная точка
text(6,0, "M*(6,0)", col = 1, adj = c(-.1, -.3))
#install.packages(lpSolveAPI)
library(lpSolveAPI) #обращение к библиотеке
M<-make.lp(ncol=2)  # Объявление количества неотриц. переменных в M
name.lp(M,"GEOM-19") # Название модели
colnames(M)<-c("x1","x2") # Названия переменных в модели
lp.control(M,sense="min")$sense # объявление задачи на минимум
set.objfn(M,c(1/4,1)) # Коэффициенты в целевой функции
add.constraint(M,c(1,3),">=", 6) #Задается  ограничение A
add.constraint(M,c(4,1),">=", 8) #Задается  ограничение B
```

```
add.constraint(M,c(3,2),">=", 11) #Задается  ограничение C
rownames(M)<-c("A","B","C")      # Обозначаются ограничения модели
#set.bounds(M,lower=c(-Inf,-Inf),upper=c(Inf,Inf))# Границы для перем.
M   # Построенная модель
```

```
Model name: GEOM-39
            x1    x2
Minimize  0.25     1
A            1     3  >=   6
B            4     1  >=   8
C            3     2  >=  11
Kind       Std   Std
Type      Real  Real
Upper      Inf   Inf
Lower        0     0
```

```
solve.lpExtPtr(M) # Решение задачи
x1_opt<-get.variables(M)[1]
x2_opt<-get.variables(M)[2]
F_min<-get.objective(M)
x1_opt;x2_opt      # Оптимальный план
```

```
[1] 6
[1] 0
```

```
F_min              # Оптимальное значение целевой функции
```

```
[1] 1.5
```

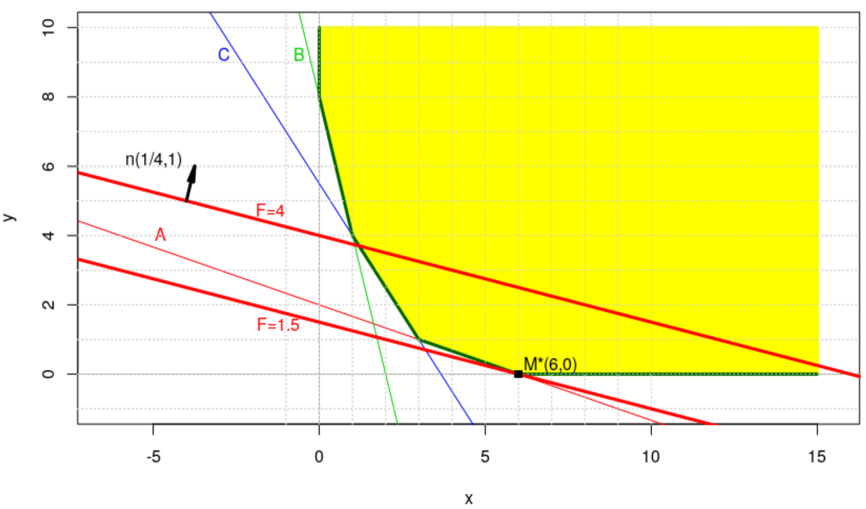

Рис. 9. Геометрический метод решения ЗЛП. Вариант 39.

Ответ. $M^*(0;6)$; $F_{min} = F(M^*) = 1,5$.

Задание 6. Задача об оптимальном использовании ресурсов [1,2 – гл. 3, 4]

Для производства трех видов изделий (A, B, C) используются ресурсы типа I и II, причем закупки ресурсов ограничены возможностями поставщиков. Нормы расхода ресурсов и их запасы приведены в таблице.

1. Постройте математическую модель задачи.
2. Определите такой план производства, при котором стоимость произведенного товара из имеющихся ресурсов является наибольшей.
3. Постройте задачу, двойственную к данной.
4. Найдите решение двойственной задачи. Поясните смысл двойственных переменных.
5. Проверьте результаты вычислений на компьютере.

Ресурсы	Изделия			Запасы
	A	B	C	
I	1	3	a	3000
II	6	5	2	3320
Цена	$6b+12$	$5b+22$	c	
План	x_1	x_2	x_3	

Данные к заданию 6:

Номер варианта	Параметры			Номер варианта	Параметры		
	a	b	c		a	b	c
1	2	1	17	21	3	3	26
2	2	2	19	22	3	4	26
3	2	3	21	23	4	1	25
4	2	4	23	24	4	1	27
5	3	1	21	25	4	2	26
6	3	1	22	26	4	2	27
7	3	2	23	27	4	3	28
8	3	2	24	28	4	3	30
9	3	2	25	29	4	4	30
10	3	3	25	30	4	4	32
11	2	1	17	31	3	3	26
12	2	2	19	32	3	4	26
13	2	3	21	33	4	1	25
14	2	4	23	34	4	1	27
15	3	1	21	35	4	2	26
16	3	1	22	36	4	2	27
17	3	2	23	37	4	3	28
18	3	2	24	38	4	3	30
19	3	2	25	39	4	4	30
20	3	3	25	40	4	4	32

Фрагменты программы с использованием пакета *linprog* и результатов исследования задачи об оптимальном использовании ресурсов

```
# Вариант 40
#F(x1, x2,x3) = 36x1 + 42x2+32x3 → max
#x1 + 3x2+4x3<= 3000,
#6x1 + 5x2+2x4<= 3320,
#x1,x2,x3 >=0.
#install.packages(linprog)
library(linprog) #обращение к библиотеке
A1<-matrix(c(1,3,4,6,5,2),nrow=2,ncol=3,byrow=TRUE) # матрица
 ↪ коэффициентов
b1<-c(3000,3320) #правые части ограничений
c1<-c(36,42,32) #коэффициенты перед переменными в целевой функции
#решение симплекс-методом
OPT1<-solveLP(c1,b1,A1,maximum=TRUE,const.dir = rep( "<=", length( b1 )
 ↪ ))
OPT1
```

```
Results of Linear Programming / Linear Optimization
Objective function (Maximum): 33360
Iterations in phase 1: 0
Iterations in phase 2: 2
Solution
  opt
1   0
2 520
3 360
Basic Variables
  opt
2 520
3 360
Constraints
  actual dir bvec free    dual dual.reg
1   3000  <= 3000    0 5.42857     1008
2   3320  <= 3320    0 5.14286     1820
All Variables (including slack variables)
    opt cvec   min.c    max.c      marg marg.reg
1     0   36    -Inf 36.28571 -0.285714  330.909
2   520   42 41.8182 80.00000        NA       NA
3   360   32 16.8000 32.30769        NA       NA
S 1   0    0    -Inf  5.42857 -5.428571 1008.000
S 2   0    0    -Inf  5.14286 -5.142857 1820.000
```

Фрагменты программы с использованием пакета *lpSolveAPI* и результатов исследования задачи об оптимальном использовании ресурсов

```
# Вариант 40
#F(x1, x2,x3) = 36x1 + 42x2+32x3 → max
#    x1 + 3x2+4x3<= 3000,
#    6x1 + 5x2+2x3<= 3320,
#    x1,x2,x3 >=0.
#install.packages(lpSolveAPI)
#library(lpSolveAPI) #обращение к библиотеке
M<-make.lp(ncol=3)  # Объявление количества неотрицательных переменных в M
name.lp(M,"ZPP-40") # Название модели
colnames(M)<-c("x1","x2","x3") # Названия переменных в модели
lp.control(M,sense="max")$sense # объявление задачи на максимум
set.objfn(M,c(36,42,32)) # Коэффициенты в целевой функции
add.constraint(M,c(1,3,4),"<=", 3000) #Задается первое ограничение
add.constraint(M,c(6,5,2),"<=", 3320) #Задается второе ограничение
rownames(M)<-c("A","B")     # Обозначаются ограничения модели
M   # Построенная модель
```

```
Model name:  ZPP-40
            x1    x2    x3
Maximize    36    42    32
A            1     3     4   <=   3000
B            6     5     2   <=   3320
Kind       Std   Std   Std
Type      Real  Real  Real
Upper      Inf   Inf   Inf
Lower        0     0     0
```

```
solve.lpExtPtr(M)
get.variables(M) # Оптимальный план
get.objective(M) # Оптимальное значение целевой функции

x1_opt<-get.variables(M)[1]
x2_opt<-get.variables(M)[2]
x3_opt<-get.variables(M)[3]
F_max<-get.objective(M)
x1_opt;x2_opt;x3_opt;F_max
```

```
[1] 0
[1] 520
[1] 360
[1] 33360
```

```r
# Анализ оптимального решения
# оценка дефицита ресурсов
b<-get.constr.value(M);b  # Заданные ограничения на ресурсы
```

```
[1] 3000 3320
```

```r
b_opt<-get.constraints(M);b_opt # Реальный расход ресурсов
```

```
[1] 3000 3320
```

```r
round(abs(b-b_opt),10)# Дефицит (исчерпанность ресурсов)
```

```
[1] 0 0
```

```r
# Оценка устойчивости коэффициентов целевой функции
# Минимальные и максимальные значения коэффициентов
# в целевой функции, сохраняющие оптимальность
min<-get.sensitivity.obj(M)$objfrom;min
```

```
[1] -1.000000e+30  4.181818e+01  1.680000e+01
```

```r
max<-get.sensitivity.obj(M)$objtill;max
```

```
[1] 36.28571 80.00000 32.30769
```

```r
# Диапозоны коэффициентов целевой функции, сохраняющие оптимальность:
cbind(min,max) # I-я строка - это диапозон устойчивости для i-го коэффициента
```

```
               min      max
[1,] -1.000000e+30 36.28571
[2,]  4.181818e+01 80.00000
[3,]  1.680000e+01 32.30769
```

```r
# Решение двойственной к M задачи
get.sensitivity.rhs(M)$duals # Оптимальный план двойственной задачи (y1,y2)
# Минимальные и максимальные значения коэффициентов в целевой функции,
# сохраняющие оптимальность
min.dual<-get.sensitivity.rhs(M)$dualsfrom;min.dual
```

```
[1]  1.992000e+03  1.500000e+03 -3.876923e+02 -1.000000e+30 -1.000000e+30
```

```
max.dual<-get.sensitivity.rhs(M)$dualstill;max.dual
```

```
[1] 6.640000e+03 5.000000e+03 3.309091e+02 1.000000e+30 1.000000e+30
```

```
# Диапозоны коэффициентов целевой функции, сохраняющие оптимальность
cbind(min.dual,max.dual) # i-я строка - диапозон для i-го коэффициента
```

```
          min.dual     max.dual
[1,]  1.992000e+03 6.640000e+03
[2,]  1.500000e+03 5.000000e+03
[3,] -3.876923e+02 3.309091e+02
[4,] -1.000000e+30 1.000000e+30
[5,] -1.000000e+30 1.000000e+30
```

Ответ. $X^* = (0; 520; 360)$; $F_{max} = F(X^*) = 33360$.

Задание 7. Транспортная задача [1,2 – гл. 6]

Дана транспортная задача: A_1, A_2, A_3 – поставщики с запасами a_1, a_2, a_3 однородного груза, B_1, B_2, B_3, B_4, B_5 – потребители с потребностями b_1, b_2, b_3, b_4, b_5. Матрица тарифов $C = (c_{ij})$, где $i = 1, 2, 3$ и $j = 1, 2, 3, 4, 5$, содержит стоимости перевозки единицы груза из пункта A_i в пункт B_j.

Требуется найти минимальный по стоимости план перевозки груза от поставщиков к потребителям такой, чтобы был вывезен весь груз и все потребности были удовлетворены.

1. Убедитесь, что транспортная задача закрытого типа.
2. Найдите какой-либо допустимый план перевозки груза (можно использовать методы северо-западного угла, минимальной стоимости или Фогеля).
3. Найдите оптимальный план перевозки груза (план минимальной стоимости) методом потенциалов.
4. Проверьте результаты вычислений на компьютере.

Данные к заданию 7:

Вариант	Матрица C	Запасы	Потребности
1.	$\begin{pmatrix} 8 & 14 & 6 & 5 & 4 \\ 13 & 12 & 9 & 13 & 7 \\ 19 & 9 & 15 & 17 & 23 \end{pmatrix}$	$a_1 = 250,$ $a_2 = 360,$ $a_3 = 280,$	$b_1 = 140,$ $b_2 = 80,$ $b_3 = 240,$ $b_4 = 310,$ $b_5 = 120.$
2.	$\begin{pmatrix} 12 & 11 & 6 & 10 & 4 \\ 15 & 12 & 8 & 13 & 7 \\ 5 & 9 & 15 & 17 & 22 \end{pmatrix}$	$a_1 = 230,$ $a_2 = 380,$ $a_3 = 250,$	$b_1 = 140,$ $b_2 = 100,$ $b_3 = 210,$ $b_4 = 270,$ $b_5 = 140.$
3.	$\begin{pmatrix} 11 & 13 & 8 & 10 & 4 \\ 13 & 11 & 12 & 9 & 5 \\ 5 & 9 & 16 & 17 & 27 \end{pmatrix}$	$a_1 = 280,$ $a_2 = 370,$ $a_3 = 240,$	$b_1 = 140,$ $b_2 = 120,$ $b_3 = 210,$ $b_4 = 270,$ $b_5 = 150.$
4.	$\begin{pmatrix} 10 & 14 & 8 & 10 & 4 \\ 12 & 11 & 12 & 13 & 5 \\ 7 & 9 & 16 & 17 & 28 \end{pmatrix}$	$a_1 = 280,$ $a_2 = 370,$ $a_3 = 240,$	$b_1 = 140,$ $b_2 = 120,$ $b_3 = 210,$ $b_4 = 270,$ $b_5 = 150.$
5.	$\begin{pmatrix} 5 & 7 & 13 & 10 & 8 \\ 6 & 15 & 12 & 9 & 4 \\ 14 & 11 & 16 & 17 & 28 \end{pmatrix}$	$a_1 = 290,$ $a_2 = 370,$ $a_3 = 240,$	$b_1 = 150,$ $b_2 = 120,$ $b_3 = 210,$ $b_4 = 270,$ $b_5 = 150.$
6.	$\begin{pmatrix} 9 & 14 & 6 & 5 & 4 \\ 13 & 15 & 12 & 3 & 7 \\ 17 & 10 & 11 & 8 & 27 \end{pmatrix}$	$a_1 = 270,$ $a_2 = 340,$ $a_3 = 290,$	$b_1 = 140,$ $b_2 = 80,$ $b_3 = 240,$ $b_4 = 310,$ $b_5 = 120.$

Вариант	Матрица C	Запасы	Потребности
7.	$\begin{pmatrix} 10 & 3 & 6 & 10 & 4 \\ 16 & 12 & 8 & 13 & 7 \\ 5 & 9 & 15 & 18 & 22 \end{pmatrix}$	$a_1 = 250$, $a_2 = 400$, $a_3 = 280$,	$b_1 = 160$, $b_2 = 110$, $b_3 = 220$, $b_4 = 270$, $b_5 = 170$.
8.	$\begin{pmatrix} 9 & 13 & 8 & 10 & 4 \\ 12 & 11 & 15 & 9 & 5 \\ 6 & 7 & 16 & 17 & 27 \end{pmatrix}$	$a_1 = 310$, $a_2 = 350$, $a_3 = 280$,	$b_1 = 160$, $b_2 = 130$, $b_3 = 220$, $b_4 = 280$, $b_5 = 150$.
9.	$\begin{pmatrix} 10 & 18 & 8 & 10 & 4 \\ 15 & 11 & 12 & 13 & 5 \\ 6 & 9 & 16 & 17 & 21 \end{pmatrix}$	$a_1 = 250$, $a_2 = 390$, $a_3 = 270$,	$b_1 = 130$, $b_2 = 150$, $b_3 = 210$, $b_4 = 270$, $b_5 = 150$.
10.	$\begin{pmatrix} 18 & 7 & 13 & 10 & 8 \\ 6 & 15 & 12 & 9 & 4 \\ 14 & 11 & 16 & 17 & 25 \end{pmatrix}$	$a_1 = 320$, $a_2 = 290$, $a_3 = 330$,	$b_1 = 160$, $b_2 = 120$, $b_3 = 210$, $b_4 = 270$, $b_5 = 180$.
11.	$\begin{pmatrix} 7 & 14 & 6 & 5 & 4 \\ 13 & 12 & 8 & 13 & 7 \\ 18 & 9 & 15 & 17 & 26 \end{pmatrix}$	$a_1 = 280$, $a_2 = 370$, $a_3 = 240$,	$b_1 = 140$, $b_2 = 80$, $b_3 = 240$, $b_4 = 310$, $b_5 = 120$.
12.	$\begin{pmatrix} 12 & 10 & 6 & 10 & 4 \\ 13 & 12 & 5 & 15 & 7 \\ 8 & 9 & 15 & 17 & 26 \end{pmatrix}$	$a_1 = 270$, $a_2 = 390$, $a_3 = 250$,	$b_1 = 160$, $b_2 = 110$, $b_3 = 210$, $b_4 = 280$, $b_5 = 150$.
13.	$\begin{pmatrix} 13 & 10 & 8 & 10 & 4 \\ 11 & 7 & 12 & 9 & 5 \\ 5 & 9 & 16 & 17 & 27 \end{pmatrix}$	$a_1 = 290$, $a_2 = 370$, $a_3 = 260$,	$b_1 = 140$, $b_2 = 130$, $b_3 = 210$, $b_4 = 270$, $b_5 = 170$.
14.	$\begin{pmatrix} 10 & 14 & 8 & 10 & 6 \\ 15 & 11 & 12 & 13 & 5 \\ 7 & 9 & 16 & 17 & 21 \end{pmatrix}$	$a_1 = 300$, $a_2 = 320$, $a_3 = 290$,	$b_1 = 130$, $b_2 = 130$, $b_3 = 210$, $b_4 = 270$, $b_5 = 170$.
15.	$\begin{pmatrix} 16 & 7 & 13 & 10 & 22 \\ 6 & 15 & 12 & 9 & 4 \\ 14 & 11 & 5 & 17 & 8 \end{pmatrix}$	$a_1 = 280$, $a_2 = 370$, $a_3 = 240$,	$b_1 = 150$, $b_2 = 120$, $b_3 = 210$, $b_4 = 260$, $b_5 = 150$.

Вариант	Матрица C	Запасы	Потребности
16.	$\begin{pmatrix} 9 & 14 & 6 & 5 & 14 \\ 3 & 17 & 12 & 13 & 7 \\ 15 & 10 & 11 & 8 & 21 \end{pmatrix}$	$a_1 = 280$, $a_2 = 390$, $a_3 = 250$,	$b_1 = 170$, $b_2 = 80$, $b_3 = 240$, $b_4 = 310$, $b_5 = 120$.
17.	$\begin{pmatrix} 10 & 3 & 6 & 11 & 4 \\ 17 & 12 & 8 & 13 & 7 \\ 18 & 9 & 15 & 5 & 24 \end{pmatrix}$	$a_1 = 250$, $a_2 = 370$, $a_3 = 280$,	$b_1 = 160$, $b_2 = 110$, $b_3 = 190$, $b_4 = 270$, $b_5 = 170$.
18.	$\begin{pmatrix} 18 & 13 & 8 & 10 & 14 \\ 12 & 11 & 15 & 9 & 5 \\ 6 & 7 & 16 & 17 & 26 \end{pmatrix}$	$a_1 = 310$, $a_2 = 320$, $a_3 = 280$,	$b_1 = 160$, $b_2 = 130$, $b_3 = 220$, $b_4 = 250$, $b_5 = 150$.
19.	$\begin{pmatrix} 9 & 17 & 8 & 10 & 4 \\ 16 & 11 & 12 & 13 & 5 \\ 6 & 9 & 16 & 18 & 21 \end{pmatrix}$	$a_1 = 280$, $a_2 = 390$, $a_3 = 270$,	$b_1 = 160$, $b_2 = 150$, $b_3 = 210$, $b_4 = 270$, $b_5 = 150$.
20.	$\begin{pmatrix} 13 & 8 & 15 & 10 & 7 \\ 6 & 18 & 12 & 9 & 4 \\ 14 & 11 & 16 & 17 & 25 \end{pmatrix}$	$a_1 = 320$, $a_2 = 250$, $a_3 = 330$,	$b_1 = 160$, $b_2 = 120$, $b_3 = 210$, $b_4 = 230$, $b_5 = 180$.
21.	$\begin{pmatrix} 11 & 14 & 6 & 5 & 4 \\ 13 & 12 & 9 & 13 & 7 \\ 23 & 9 & 15 & 17 & 18 \end{pmatrix}$	$a_1 = 250$, $a_2 = 360$, $a_3 = 290$,	$b_1 = 140$, $b_2 = 90$, $b_3 = 240$, $b_4 = 310$, $b_5 = 120$.
22.	$\begin{pmatrix} 12 & 11 & 6 & 10 & 4 \\ 16 & 12 & 8 & 13 & 5 \\ 7 & 9 & 16 & 17 & 22 \end{pmatrix}$	$a_1 = 270$, $a_2 = 380$, $a_3 = 260$,	$b_1 = 170$, $b_2 = 120$, $b_3 = 210$, $b_4 = 270$, $b_5 = 140$.
23.	$\begin{pmatrix} 16 & 14 & 8 & 10 & 4 \\ 13 & 11 & 12 & 6 & 5 \\ 5 & 9 & 16 & 17 & 27 \end{pmatrix}$	$a_1 = 280$, $a_2 = 330$, $a_3 = 290$,	$b_1 = 150$, $b_2 = 120$, $b_3 = 210$, $b_4 = 270$, $b_5 = 150$.
24.	$\begin{pmatrix} 5 & 14 & 8 & 11 & 4 \\ 15 & 10 & 12 & 13 & 10 \\ 7 & 9 & 16 & 17 & 23 \end{pmatrix}$	$a_1 = 290$, $a_2 = 370$, $a_3 = 250$,	$b_1 = 140$, $b_2 = 130$, $b_3 = 210$, $b_4 = 270$, $b_5 = 160$.

Вариант	Матрица C	Запасы	Потребности
25.	$\begin{pmatrix} 8 & 7 & 14 & 10 & 5 \\ 6 & 13 & 12 & 9 & 4 \\ 27 & 11 & 16 & 17 & 13 \end{pmatrix}$	$a_1 = 290$, $a_2 = 370$, $a_3 = 270$,	$b_1 = 180$, $b_2 = 120$, $b_3 = 210$, $b_4 = 270$, $b_5 = 150$.
26.	$\begin{pmatrix} 9 & 12 & 6 & 5 & 4 \\ 11 & 15 & 14 & 13 & 7 \\ 17 & 10 & 3 & 8 & 25 \end{pmatrix}$	$a_1 = 270$, $a_2 = 350$, $a_3 = 290$,	$b_1 = 80$, $b_2 = 180$, $b_3 = 210$, $b_4 = 310$, $b_5 = 130$.
27.	$\begin{pmatrix} 4 & 3 & 6 & 10 & 11 \\ 17 & 7 & 8 & 13 & 7 \\ 5 & 9 & 15 & 19 & 22 \end{pmatrix}$	$a_1 = 250$, $a_2 = 400$, $a_3 = 220$,	$b_1 = 160$, $b_2 = 100$, $b_3 = 190$, $b_4 = 270$, $b_5 = 150$.
28.	$\begin{pmatrix} 19 & 13 & 8 & 10 & 14 \\ 3 & 11 & 15 & 9 & 5 \\ 6 & 7 & 16 & 17 & 4 \end{pmatrix}$	$a_1 = 310$, $a_2 = 300$, $a_3 = 280$,	$b_1 = 110$, $b_2 = 130$, $b_3 = 220$, $b_4 = 280$, $b_5 = 150$.
29.	$\begin{pmatrix} 5 & 18 & 8 & 10 & 4 \\ 16 & 11 & 12 & 13 & 7 \\ 6 & 9 & 16 & 17 & 21 \end{pmatrix}$	$a_1 = 260$, $a_2 = 390$, $a_3 = 270$,	$b_1 = 130$, $b_2 = 150$, $b_3 = 210$, $b_4 = 270$, $b_5 = 160$.
30.	$\begin{pmatrix} 18 & 7 & 13 & 10 & 26 \\ 6 & 17 & 12 & 9 & 4 \\ 24 & 11 & 15 & 17 & 8 \end{pmatrix}$	$a_1 = 320$, $a_2 = 250$, $a_3 = 330$,	$b_1 = 170$, $b_2 = 120$, $b_3 = 210$, $b_4 = 220$, $b_5 = 180$.
31.	$\begin{pmatrix} 9 & 4 & 6 & 5 & 6 \\ 17 & 11 & 15 & 3 & 7 \\ 20 & 9 & 15 & 7 & 25 \end{pmatrix}$	$a_1 = 270$, $a_2 = 390$, $a_3 = 290$,	$b_1 = 150$, $b_2 = 100$, $b_3 = 250$, $b_4 = 340$, $b_5 = 110$.

Фрагменты программы с использованием пакета *lpSolve*
и результатов исследования транспортной задачи

```
# Транспортная задача. Вариант 31
# В Excel вводим матрицу тарифов и помещаем в буфер обмена
#install.packages(lpSolve)
library(lpSolve) # Активация библиотеки пакета lpSolve
# Ввод матрицы тарифов
# Чтение из буфера обмена
#Data<-read.table("clipboard",h=FALSE,dec=",",sep="")
#P<-as.matrix(Data);P # Объявление таблицы Data матрицей в R
P<-matrix(c(9,4,6,5,6,17,11,15,3,7,20,9,15,7,25),byrow=TRUE,nrow=3,ncol=5)
P
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    9    4    6    5    6
[2,]   17   11   15    3    7
[3,]   20    9   15    7   25
```

```
# Введение ограничений
SignA<-c("==","==","==")
SignB<-c(rep("==",5))
SumA<-c(270,390,290) # Запасы по складам
SumB<-c(150,100,250,340,110) # Потребности
# Модель  транспортной задачи
M.Tr<-lp.transport(cost.mat=P,direction="min",
                   row.signs=SignA,row.rhs=SumA,
                   col.signs=SignB,col.rhs=SumB)
M.Tr
```

```
Success: the objective function is 6950
```

```
M.Tr$status
```

```
[1] 0
```

```
M.Tr$solution   # Оптимальный план перевозки груза
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]  150    0  120    0    0
[2,]    0    0    0  280  110
[3,]    0  100  130   60    0
```

23
```
M.Tr$objval # Оптимальное значение целевой функции
```

```
[1] 6950
```

24
```
str(M.Tr)
```

```
List of 20
 $ direction     : int 0
 $ rcount        : int 3
 $ ccount        : int 5
 $ costs         : num [1:16] 0 9 17 20 4 11 9 6 15 15 ...
 $ rsigns        : int [1:3] 3 3 3
 $ rrhs          : num [1:3] 270 390 290
 $ csigns        : int [1:5] 3 3 3 3 3
 $ crhs          : num [1:5] 150 100 250 340 110
 $ objval        : num 6950
 $ int.count     : int 15
 . . . . .
```

25
```
M.Tr$solution # Оптимальный план перевозки груза:
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]  150    0  120    0    0
[2,]    0    0    0  280  110
[3,]    0  100  130   60    0
```

26
```
M.Tr$objval  # Оптимальная стоимость перевозки:
```

```
[1] 6950
```

Ответ. $X^* = \begin{pmatrix} 150 & 0 & 120 & 0 & 0 \\ 0 & 0 & 0 & 280 & 110 \\ 0 & 100 & 130 & 60 & 0 \end{pmatrix}$; $F_{min} = F(X^*) = 6950.$

Задание 8. Задача теории игр [1,2 – гл. 13]
Игра задана платежной матрицей A.
1. Составьте пару двойственных задач, соответствующую игрокам.
2. Найдите оптимальные стратегии игроков и цену игры.
3. Проверьте результаты вычислений на компьютере.

Данные к заданию 8:

Вариант	Матрица A	Вариант	Матрица A
1.	$\begin{pmatrix} 4 & 5 & 7 \\ 7 & 4 & 5 \\ 5 & 7 & 4 \end{pmatrix}$	2.	$\begin{pmatrix} 3 & 5 & 6 \\ 6 & 3 & 5 \\ 5 & 6 & 3 \end{pmatrix}$
3.	$\begin{pmatrix} 2 & 3 & 5 \\ 3 & 5 & 2 \\ 5 & 2 & 3 \end{pmatrix}$	4.	$\begin{pmatrix} 1 & 2 & 6 \\ 6 & 1 & 2 \\ 2 & 6 & 1 \end{pmatrix}$
5.	$\begin{pmatrix} 0 & 2 & 5 \\ 7 & 1 & 3 \\ 2 & 5 & 0 \end{pmatrix}$	6.	$\begin{pmatrix} 7 & 3 & 4 \\ 4 & 6 & 3 \\ 3 & 4 & 7 \end{pmatrix}$
7.	$\begin{pmatrix} 3 & 1 & 5 \\ 5 & 3 & 1 \\ 1 & 5 & 3 \end{pmatrix}$	8.	$\begin{pmatrix} 1 & 4 & 5 \\ 5 & 1 & 4 \\ 4 & 5 & 1 \end{pmatrix}$
9.	$\begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 2 & 5 \end{pmatrix}$	10.	$\begin{pmatrix} 1 & 4 & 2 \\ 6 & 1 & 4 \\ 4 & 6 & 1 \end{pmatrix}$
11.	$\begin{pmatrix} 4 & 0 & 2 \\ 3 & 5 & 1 \\ 0 & 2 & 4 \end{pmatrix}$	12.	$\begin{pmatrix} 3 & 2 & 4 \\ 4 & 3 & 2 \\ 2 & 4 & 3 \end{pmatrix}$
13.	$\begin{pmatrix} 2 & 3 & 1 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$	14.	$\begin{pmatrix} 6 & 3 & 2 \\ 2 & 6 & 3 \\ 3 & 2 & 6 \end{pmatrix}$
15.	$\begin{pmatrix} 1 & 2 & 1 \\ 1 & 1 & 3 \\ 3 & 1 & 2 \end{pmatrix}$	16.	$\begin{pmatrix} 4 & 3 & 5 \\ 5 & 4 & 3 \\ 3 & 5 & 4 \end{pmatrix}$
17.	$\begin{pmatrix} 2 & 4 & 0 \\ 1 & 3 & 5 \\ 4 & 0 & 2 \end{pmatrix}$	18.	$\begin{pmatrix} 2 & 3 & 5 \\ 3 & 5 & 2 \\ 5 & 2 & 3 \end{pmatrix}$
19.	$\begin{pmatrix} 3 & 5 & 1 \\ 4 & 0 & 2 \\ 0 & 2 & 4 \end{pmatrix}$	20.	$\begin{pmatrix} 1 & 3 & 4 \\ 4 & 1 & 3 \\ 3 & 4 & 1 \end{pmatrix}$
21.	$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{pmatrix}$	22.	$\begin{pmatrix} 2 & 4 & 5 \\ 5 & 2 & 4 \\ 4 & 5 & 2 \end{pmatrix}$
23.	$\begin{pmatrix} 1 & 1 & 2 \\ 3 & 1 & 1 \\ 2 & 3 & 1 \end{pmatrix}$	24.	$\begin{pmatrix} 1 & 2 & 5 \\ 5 & 1 & 2 \\ 2 & 5 & 1 \end{pmatrix}$
25.	$\begin{pmatrix} 0 & 2 & 4 \\ 5 & 1 & 3 \\ 2 & 4 & 0 \end{pmatrix}$	26.	$\begin{pmatrix} 6 & 3 & 4 \\ 4 & 6 & 3 \\ 3 & 4 & 6 \end{pmatrix}$

Вариант	Матрица A	Вариант	Матрица A
27.	$\begin{pmatrix} 3 & 1 & 2 \\ 2 & 3 & 1 \\ 1 & 2 & 3 \end{pmatrix}$	28.	$\begin{pmatrix} 1 & 4 & 2 \\ 2 & 1 & 4 \\ 4 & 2 & 1 \end{pmatrix}$
29.	$\begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 2 & 3 \end{pmatrix}$	30.	$\begin{pmatrix} 1 & 4 & 6 \\ 6 & 1 & 4 \\ 4 & 6 & 1 \end{pmatrix}$
31.	$\begin{pmatrix} 1 & 2 & 3 \\ 5 & 3 & 4 \\ 1 & 4 & 0 \end{pmatrix}$		

Фрагменты программы с использованием пакета *lpSolveAPI*
и результатов исследования матричной игры

```
#Вариант 31. Решите матричную игру
# A=(1,2,3//5,3,4//1,4,0)
#F(w1, w2,w3) = w1+w2+w3 → max
#   w1 + 2w2+3w3<= 1,
#   5w1 + 3w2+4w3<= 1,
#   w1 + 4w2<= 1,
#    w1,w2,w3 >=0.
#install.packages(lpSolveAPI)
library(lpSolveAPI) #обращение к библиотеке
M<-make.lp(ncol=3)  # Объявление количества неотрицательных переменных в M
name.lp(M,"Matr-Game") # Название модели
colnames(M)<-c("w1","w2","w3") # Названия переменных в модели
lp.control(M,sense="max")$sense # объявление задачи на максимум
set.objfn(M,c(1,1,1)) # Коэффициенты в целевой функции
add.constraint(M,c(1,2,3),"<=", 1) #Задается первое ограничение
add.constraint(M,c(5,3,4),"<=", 1) #Задается второе ограничение
add.constraint(M,c(1,4,0),"<=", 1) #Задается третье ограничение
rownames(M)<-c("A","B","C")     # Обозначаются ограничения модели
M   # Построенная модель
```

```
Model name: Matr-Game
            w1    w2    w3
Maximize     1     1     1
A            1     2     3  <=  1
B            5     3     4  <=  1
C            1     4     0  <=  1
Kind       Std   Std   Std
Type      Real  Real  Real
Upper      Inf   Inf   Inf
Lower        0     0     0
```

```
solve.lpExtPtr(M)
get.variables(M) # Оптимальный план
get.objective(M) # Оптимальное значение целевой функции
```

```
[1] 0.0000 0.2500 0.0625
[1] 0.3125
```

```
w1_opt<-get.variables(M)[1]
w2_opt<-get.variables(M)[2]
w3_opt<-get.variables(M)[3]
F_max<-get.objective(M)
w1_opt;w2_opt;w3_opt
F_max
```

```
[1] 0
[1] 0.25
[1] 0.0625
[1] 0.3125
```

```
# Анализ оптимального решения
# оценка выполнения ограничений
b<-get.constr.value(M);b  # Заданные ограничения
b_opt<-get.constraints(M);b_opt # Реальный расход
round(abs(b-b_opt),10)# Дефицит
```

```
[1] 1 1 1
[1] 0.6875 1.0000 1.0000
[1] 0.3125 0.0000 0.0000
```

```
# Оценка устойчивости коэффициентов целевой функции
min<-get.sensitivity.obj(M)$objfrom;min
max<-get.sensitivity.obj(M)$objtill;max
```

```
[1] -1.000000e+30 7.500000e-01 7.058824e-01
[1] 1.312500e+00 1.000000e+30 1.333333e+00
```

```
# Диапозоны коэффициентов целевой функции, сохраняющие оптимальность
cbind(min,max)
```

```
               min          max
[1,] -1.000000e+30 1.312500e+00
[2,]  7.500000e-01 1.000000e+30
[3,]  7.058824e-01 1.333333e+00
```

```
#Оптимальные стратегии второго игрока и цена игры
Sumw<-w1_opt+w2_opt+w3_opt
y1<-w1_opt/Sumw
y2<-w2_opt/Sumw
y3<-w3_opt/Sumw
Nu_A<-1/Sumw
y1;y2;y3
Nu_A
```

```
[1] 0
[1] 0.8
[1] 0.2
[1] 3.2
```

```
# Решение двойственной задачи (задачи для первого игрока)
u1_opt<-get.sensitivity.rhs(M)$duals[1]
u2_opt<-get.sensitivity.rhs(M)$duals[2]
u3_opt<-get.sensitivity.rhs(M)$duals[3]
u1_opt;u2_opt;u3_opt
```

```
[1] 0
[1] 0.25
[1] 0.0625
```

```
#Оптимальные стратегии первого игрока
Sumu<-u1_opt+u2_opt+u3_opt
x1<-u1_opt/Sumu
x2<-u2_opt/Sumu
x3<-u3_opt/Sumu
x1;x2;x3
```

```
[1] 0
[1] 0.8
[1] 0.2
```

Ответ. $X^* = (0; 0,8; 0,2)$ – оптимальная стратегия I игрока, $Y^* = (0; 0,8; 0,2)$ – оптимальная стратегия II игрока, $\nu_A = 3,2$ – цена игры.

Литература

1. Методы оптимальных решений в экономике и финансах: учебник / Под ред. В.М. Гончаренко, В.Ю. Попова. – М.: Кнорус, 2014. – 400с.

2. Методы оптимальных решений в экономике и финансах. Практикум: Учебное пособие / Под ред. В.М. Гончаренко, В.Ю. Попова. – М.: Кнорус, 2016.

3. Александрова И.А. Методы оптимальных решений. Руководство к решению задач: учебн. пособ. / И.А. Александрова, В.М. Гончаренко. – М.: Финуниверситет, 2012.

4. Соловьев В.И. Методы оптимальных решений: учебн. пособ. / Финуниверситет. – М.: Финуниверситет, 2012.

5. Винюков И.А. Линейная алгебра: учебн. пособ. Ч. 4. Линейное программирование / И.А. Винюков, В.Ю. Попов, С.В. Пчелинцев; Под ред. В.Б. Гисина, С.В. Пчелинцева. – М.: Финуниверситет, 2013.

6. Кабаков. Р.И. R в действии. Анализ и визуализация данных в программе R / Р.И. Кабаков; пер. с англ. П. Волковой. – М.: ДМК Пресс, 2014. – 588 с.

7. Зададаев С.А. Математика на языке R: учебник / Финансовый университет при Правительстве РФ. – М.: Прометей, 2018. – 324с.

8. Орлова И.В., Бич М.Г. Экономико-математическое моделирование: практическое пособие по решению задач в Excel и R / И.В. Орлова, М.Г. Бич. – 3-е изд., испр. и доп. – М.: Вузовский учебник: ИНФРА-М, 2018. – 190 с.