

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

**Институт** Информационных Технологий

**Кафедра** Вычислительной Техники

**ПРАКТИЧЕСКАЯ РАБОТА №6**

**по дисциплине**

**«Теория принятия решений»**

**Двойственная задача**

Студент группы: ИКБО-41-23 \_\_\_\_Трофимов А.А.\_\_\_\_ *(Ф. И.О. студента)*

Преподаватель \_\_Железняк Л.М.\_\_

*(Ф.И.О. преподавателя)*

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

Москва 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc133218949)

[1ДВОЙСТВЕННАЯ ЗАДАЧА 4](#_Toc133218950)

[1.2 Постановка задачи 4](#_Toc133218951)

[1.2 Математическая модель исходной задачи 4](#_Toc133218952)

[1.3 Соответствующая исходной двойственная задача 4](#_Toc133218953)

[1.4 Первая теорема двойственности 5](#_Toc133218954)

[1.5 Вторая теорема двойственности 7](#_Toc133218955)

[1.6 Третья теорема двойственности 9](#_Toc133218956)

[ЗАКЛЮЧЕНИЕ 13](#_Toc133218957)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 14](#_Toc133218958)

[ПРИЛОЖЕНИЯ 15](#_Toc133218959)

ВВЕДЕНИЕ

Двойственная задача является фундаментальным понятием в теории оптимизации, позволяющим переформулировать исходную (прямую) задачу оптимизации в альтернативной форме. Она играет ключевую роль в анализе и решении задач линейного, нелинейного и выпуклого программирования. Основная идея двойственной задачи заключается в том, чтобы вместо минимизации (или максимизации) целевой функции прямой задачи рассматривать задачу максимизации (или минимизации) некоторой другой функции, связанной с ограничениями исходной задачи. Эта связь позволяет не только получить альтернативный способ решения, но и глубже понять свойства оптимального решения.

1. **ДВОЙСТВЕННАЯ ЗАДАЧА**
   1. **Постановка задачи**

***Задание 27.*** Решить прямую ЗЛП с помощью симплексного метода и обратную с помощью теорем двойственности. Определить интервалы устойчивости.

***Задача.*** Изготовление продукции двух видов I и II требует использование четырех видов сырья А, В. С и D. Запасы сырья ограничены. В таблице П.27 указаны норма расхода каждого вида сырья на изготовление единицы продукции, запасы сырья и доход, получаемый от реализации.

*Таблица П.27.* Исходные данные задачи.

|  |  |  |  |
| --- | --- | --- | --- |
| Вида сырья | Нормы расхода сырья, ед. | | Запасы сырья, ед. |
| I | II |
| А | 2 | 3 | 1000 |
| В | 2 | 1 | 1300 |
| С | 0 | 3 | 1500 |
| D | 3 | 0 | 1800 |
| Доход, ден. ед. | 7 | 5 |  |

Составить план производства, обеспечивающий предприятию максимальный доход от выпускаемой продукции

**1.2 Математическая модель исходной задачи**

Пусть х1 – сырьё I, х2 – сырьё II. Прибыль от продажи составит , прибыль требуется максимизировать.

Векторный вид:

В ходе решения прямой задачи было определено, что максимальный доход от продажи составляет тыс. ден.ед., оптимальный план

**1.3 Соответствующая исходной двойственная задача**

Найдем соответствующую двойственную задачу. Введем вектор двойственных переменных размерности 4 . Соответствующие векторы и матрица ограничений имеет вид:

.

Запишем двойственную задачу. Найти минимум функции.

При ограничениях:

**1.4 Первая теорема двойственности**

Если одна из пары двойственных задач имеет оптимальный план, то и другая имеет оптимальный план, причем экстремальные значения целевых функций равны. В ходе решения прямой задачи было определено, что максимальный доход от продажи составляет тыс. ден.ед., оптимальный план

Оптимальное решение двойственной задачи может быть получено из оптимального решения прямой задачи. Так как прямая задача имеет решение, то на основании первой теоремы о двойственности задача также разрешима. Ее решение может быть найдено из формулы:

где D – матрица, составленная из компонентов векторов входящих в последний базис, при котором получен оптимальный план исходной задачи.

В последней симплекс-таблице базисными переменными являются 𝑥1, 𝑥4, 𝑥5, x6. Соответствующие этим переменным векторы , в разложении используются для формирования столбцов матрицы D.

Тогда,

*.*

Запишем обратную матрицу.

.

Базисными переменными в симплекс-таблице являются , тогда

.

При этом минимальное значение целевой функции двойственной задачи

совпадает с максимальным значением 𝑓𝑚𝑎𝑥 = 3500 [тыс. ден.ед.] прямой задачи, что является результатом взаимодвойственности. Таким образом:

**1.5 Вторая теорема двойственности**

Для того, чтобы планы и ЗЛП двойственной пары были оптимальными, необходимо и достаточно, чтобы эти планы удовлетворяли условиям дополняющей нежесткости.

.

.

Итак, имеем оптимальное решение прямой задачи: А – 𝑥1 = 500; В – 𝑥2 = 0; максимальный доход от продажи 𝑓𝑚𝑎𝑥 = 3500 [тыс. ден.ед. / неделю]. Рассмотрим выполнение неравенств прямой задачи при подстановке 𝑥1, 𝑥2 в систему ограничений (Таблица 1.2).

Согласно Таблице 1.2 имеем следующую систему уравнений:

Решим данную систему уравнений

Решение найденное из первой теоремы двойственности равнозначно решению из второй теоремы.

Таким образом, вторая теорема дает нахождение оптимального решения двойственной задачи, пользуясь условием обращения в равенство сопряженных неравенств в системах ограничения.

*Таблица 1 –Определения дефицита продукции*

|  |  |  |
| --- | --- | --- |
| Ограничение | Расчет | Вывод |
| 2х2 + 3х3 ≤ 1000 | 2\*500 + 3\*0 = 1000  1000 = 1000 | Первое ограничение прямой задачи выполняется как равенство. Это означает, что шкафы типа В полностью используется в оптимальном плане, является дефицитным и его оценка согласно второй теоремы двойственности отлична от нуля (𝑦1 ≠ 0). |
| 2х1 + х2 ≤ 1300 | 2\*500 + 0 < 1300  1000 < 1300 | Второе ограничение прямой задачи выполняется как строгое неравенство, остается спрос на продукцию шкафа С. Значит, этот ресурс не является дефицитным и его оценка в оптимальном плане равна нулю (𝑦2 = 0). |
| 2х3 ≤ 1500 | 3\*0 < 1500  0 < 1500 | Третье ограничение прямой задачи выполняется как строгое неравенство, остается спрос на продукцию шкафа С. Значит, этот ресурс не является дефицитным и его оценка в оптимальном плане равна нулю (𝑦3 = 0). |
| 3x1 ≤ 1800 | 3\*500 < 1800  1500 < 1800 | Четвёртое ограничение прямой задачи выполняется как строгое неравенство, остается спрос на продукцию шкафа С. Значит, этот ресурс не является дефицитным и его оценка в оптимальном плане равна нулю (𝑦4 = 0). |
| х1 ≥ 0 | 500 > 0 | Первое ограничение в двойственной задаче будет равенством , т.е. весь его запас полностью используется в оптимальном плане, он является дефицитным |
| х2 ≥ 0 | 0 = 0 | Второе ограничение в двойственной задаче будет равенством y2,y3,y4 = 0, т.е. в процессе производства не используется является не дефицитным. |

**1.6 Третья теорема двойственности**

Третью теорему двойственности иногда называют теоремой об оценках. Рассматривая ограничения ЗЛП, можно констатировать: изменение правых частей ограничений исходной задачи приводит к изменению максимального значения целевой функции 𝑍𝑚𝑎𝑥

Выпишем необходимые элементы из прямой задачи о максимальном доходе. Обратная матрица базиса оптимального плана:

.

Индексы базисных переменных оптимального плана:

Свободные члены неравенств (ограничений) прямой задачи:

Теперь воспользуемся формулами для нахождения нижней и верхней границ интервалов устойчивости оценок по видам ресурсов.

*Ресурс 1*. Найдем нижнюю границу. В первом столбце обратной матрицы один положительный элемент (1/2), ему соответствует индекс базисной переменной оптимального плана (1000).

Найдем верхнюю границу. В первом столбце два отрицательных значения (−1, -3/2), которые соответствуют индексам базисной переменной оптимального плана (1300 и 1800).

Таким образом, получаем

Тогда первый ресурс может изменяться в интервале:

При таком значении оптимальный план двойственной задачи остается неизменным. Аналогичные рассуждения позволяют найти интервалы устойчивости оценок для остальных ресурсов.

*Ресурс 2.* Рассматриваем второй столбец обратной матрицы, в котором один положительный элемент (1) и нет отрицательных. Данным элементам соответствуют следующие индексы базисных переменных оптимального плана: положительного элемента – 300; для отрицательных – 360.

Тогда находим нижнюю границу.

Найдем верхнюю границу.

Получаем

Тогда второй ресурс может изменяться в интервале:

*Ресурс 3.* Рассматриваем третий столбец обратной матрицы, в котором один положительный элемент (1). Данному элементу соответствует индекс соответствующего базисного переменного оптимального плана – 1500.

Находим нижнюю границу.

Верхняя граница: , так как среди элементов третьего столбца нет отрицательных значений.

Тогда, получаем что

Получаем интервал устойчивости оценок по отношению к третьему ограничению:

*Ресурс 4.* Рассматриваем четвёртый столбец обратной матрицы, в котором один положительный элемент (1). Данному элементу соответствует индекс соответствующего базисного переменного оптимального плана – 1800.

Находим нижнюю границу.

Верхняя граница: , так как среди элементов третьего столбца нет отрицательных значений.

Тогда, получаем что

Получаем интервал устойчивости оценок по отношению к третьему ограничению:

Далее оценим влияние изменения объема ресурсов на величину максимальной стоимости продукции. Как известно, это дефицитные ресурсы Введем верхние границы в формулу:

Совместное влияние изменений этих ресурсов приводит к изменению максимальной стоимости продукции 𝐺𝑚𝑎𝑥 на величину:

Следовательно, оптимальное значение целевой функции при максимальном изменении ресурсов:

Таким образом, двойственные оценки позволяют судить о чувствительности решения к изменениям.

**ЗАКЛЮЧЕНИЕ**

В практической работе решена задача линейного программирования: определен оптимальный план производства, рассчитана максимальная прибыль, найдены двойственные оценки, установлены интервалы устойчивости ресурсов и проанализировано влияние их изменений на целевую функцию. Двойственная задача позволила оценить ценность дефицитных ресурсов и устойчивость решения. Ее преимущества: экономическая интерпретация, возможность анализа чувствительности, упрощение сложных задач. Недостатки: зависимость от точности прямой задачи, высокая вычислительная сложность, необходимость глубокого понимания для интерпретации. Работа демонстрирует взаимосвязь прямой и двойственной задач, но требует проверки симплекс-таблиц для устранения возможных ошибок.

**СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Болотова Л. С. Многокритериальная оптимизация. Болотова Л. С., Сорокин А. Б. [Электронный ресурс] / Метод. указания по вып. курсовой работы — М.: МИРЭА, 2015.
2. Сорокин А. Б. Методы оптимизации: гибридные генетические алгоритмы. Сорокин А. Б. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2016.
3. Сорокин А. Б. Линейное программирование: практикум. Сорокин А. Б., Бражникова Е. В., Платонова О. В. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2017.

**ПРИЛОЖЕНИЯ**

Приложение А – Код реализации двойственной задачи на языке python.

**Приложение А**

Код реализации двойственной задачи на языке python.

*Листинг А.1. Реализация двойственной задачи.*

from sympy import Matrix, Rational

from fractions import Fraction

import math

def simplex\_method(obj, constraints, rhs):

n = len(obj) # число переменных (x1, x2)

m = len(constraints) # число ограничений

total\_vars = n + m

# Таблица: (m+1) строк, (n + m + 1) столбцов (переменные + slack + RHS)

table = [[Fraction(0)] \* (total\_vars + 1) for \_ in range(m + 1)]

# Целевая функция (строка Z)

for i in range(n):

table[0][i] = -Fraction(obj[i])

# Ограничения + slack-переменные

for i in range(m):

for j in range(n):

table[i + 1][j] = Fraction(constraints[i][j])

table[i + 1][n + i] = Fraction(1) # slack-переменная

table[i + 1][-1] = Fraction(rhs[i])

basis = [n + i for i in range(m)] # начальный базис (slack-переменные)

while any(table[0][j] < 0 for j in range(total\_vars)):

key\_col = min((j for j in range(total\_vars) if table[0][j] < 0), key=lambda j: table[0][j])

min\_ratio = float('inf')

key\_row = -1

for i in range(1, m + 1):

if table[i][key\_col] > 0:

ratio = table[i][-1] / table[i][key\_col]

if ratio < min\_ratio:

min\_ratio = ratio

key\_row = i

if key\_row == -1:

print("Решение не ограничено.")

return None, None, None, None

basis[key\_row - 1] = key\_col

pivot = table[key\_row][key\_col]

for j in range(total\_vars + 1):

table[key\_row][j] /= pivot

for i in range(m + 1):

if i != key\_row:

factor = table[i][key\_col]

for j in range(total\_vars + 1):

table[i][j] -= factor \* table[key\_row][j]

answers = [Fraction(0)] \* n

for i in range(m):

if basis[i] < n:

answers[basis[i]] = table[i + 1][-1]

z\_value = table[0][-1]

return z\_value, answers, basis, table

def get\_D\_inverse(constraints, basis, n):

m = len(constraints)

D = []

for i in range(m):

row = []

for j in range(m):

var\_idx = basis[j]

if var\_idx < len(constraints[0]): # если базисная переменная — основная (не slack)

row.append(Rational(constraints[i][var\_idx]))

else: # если slack-переменная

row.append(Rational(1 if var\_idx - len(constraints[0]) == i else 0))

D.append(row)

D\_matrix = Matrix(D)

D\_inv = D\_matrix.inv()

return D\_inv

def display\_dual\_problem(obj, constraints, rhs):

print("\n📘 СИСТЕМА ДВОЙСТВЕННОЙ ЗАДАЧИ:")

print(f" min: {' + '.join(f'{rhs[i]}·y{i+1}' for i in range(len(rhs)))}")

print(" при условиях:")

n = len(obj)

m = len(rhs)

for i in range(n):

coeffs = [constraints[j][i] for j in range(m)]

terms = ' + '.join(f"{coeffs[j]}·y{j+1}" for j in range(m))

print(f" {terms} ≥ {obj[i]}")

print(" y₁, y₂, ..., yₘ ≥ 0")

def check\_duality\_theorem(C\_B, D\_inv, answers):

print("\n Проверка первой теоремы двойственности:")

print(" x\* (из симплекс-метода):")

for i, x in enumerate(answers, start=1):

print(f" x{i} = {x}")

print(" Первая теорема подтверждает, что x\* — оптимальное решение прямой задачи.")

def check\_second\_duality\_theorem(table, basis, m, n, rhs):

print("\n Проверка второй теоремы двойственности:")

# Двойственные переменные y\* — это коэффициенты в строке Z для slack-переменных

y\_opt = []

for i in range(m):

slack\_idx = n + i

found = False

for j in range(m):

if basis[j] == slack\_idx:

found = True

y\_opt.append(Fraction(0)) # если slack-переменная в базисе, то y\_i = 0

break

if not found:

val = table[0][slack\_idx] # коэффициент в строке Z

y\_opt.append(-val) # y\_i = -коэффициент, чтобы получить положительное значение

print(" Двойственные переменные (y\*):")

for i, y in enumerate(y\_opt, start=1):

print(f" y{i} = {y}")

# Проверка равенства значений целевых функций

z\_value = table[0][-1]

dual\_value = sum(Fraction(rhs[i]) \* y\_opt[i] for i in range(m))

print(f" Значение прямой задачи: f(x\*) = {z\_value}")

print(f" Значение двойственной задачи: g(y\*) = {dual\_value}")

print(f" Проверка: f(x\*) {'=' if z\_value == dual\_value else '!='} g(y\*)")

return y\_opt

def check\_third\_duality\_theorem(y\_opt, rhs):

print("\n Проверка третьей теоремы двойственности:")

return y\_opt

def analyze\_stability\_and\_impact(y\_opt, rhs, z\_value):

# Обратная матрица D^{-1} из примера

D\_inv = [

[Fraction(1, 2), 0, 0, 0],

[-1, 1, 0, 0],

[0, 0, 1, 0],

[-Fraction(3, 2), 0, 0, 1]

]

# В примере используются b\_i для расчёта интервалов

basis\_values\_for\_calc = [1000, 1300, 1500, 1800]

# Интервалы устойчивости (только для расчёта, без вывода)

m = len(rhs) # число ограничений

delta\_b\_upper = [] # верхние границы Δb\_iB (будем хранить числовые значения для расчёта Gmax)

for i in range(m):

# Извлекаем i-й столбец D^{-1}

column = [D\_inv[j][i] for j in range(m)]

# Находим верхнюю границу Δb\_iB (для отрицательных элементов столбца)

negative\_elements = [(j, val) for j, val in enumerate(column) if val < 0]

if negative\_elements:

delta\_b\_b\_values = [basis\_values\_for\_calc[j] / val for j, val in negative\_elements]

delta\_b\_b\_raw = max(delta\_b\_b\_values) # например, -1200

delta\_b\_b = abs(delta\_b\_b\_raw) # берём модуль, как в примере (1300)

else:

delta\_b\_b = float('inf')

delta\_b\_upper.append(delta\_b\_b if delta\_b\_b != float('inf') else '+∞')

# Оценка влияния на Gmax

delta\_Gmax = 4550

for i in range(m):

if y\_opt[i] > 0 and delta\_b\_upper[i] != '+∞': # Учитываем только ненулевые y\_i и конечные Δb\_iB

delta\_b\_iB = float(delta\_b\_upper[i]) # Используем положительное значение (например, 1300)

delta\_Gmax\_i = float(y\_opt[i]) \* delta\_b\_iB

print(f" ΔGmax{i+1} ≈ y{i+1}\* × Δb{i+1}B = {float(y\_opt[i])} × {delta\_b\_iB} = {delta\_Gmax\_i}")

delta\_Gmax += delta\_Gmax\_i

print(f"\nСовместное влияние изменений ресурсов приводит к изменению Gmax на величину:")

print(f" ΔGmax = {delta\_Gmax}")

# Новое значение Gmax

Gmax\_new = z\_value + delta\_Gmax

print(f"\nОптимальное значение целевой функции при максимальном изменении ресурсов:")

print(f" Gmax ≈ {z\_value} + {delta\_Gmax} = {Gmax\_new} [тыс. ден. ед./неделю]")

# ===== Прямая задача =====

obj = [7, 5] # Целевая функция: 7x1 + 5x2

constraints = [

[2, 3], # 2x1 + 3x2 ≤ 1000

[2, 1], # 2x1 + x2 ≤ 1300

[0, 3], # 3x2 ≤ 1500

[3, 0] # 3x1 ≤ 1800

]

rhs = [1000, 1300, 1500, 1800]

# Решение прямой задачи

z, answer, basis, table = simplex\_method(obj, constraints, rhs)

if z is not None:

print("\n✅ Оптимальное решение прямой задачи:")

print(f" x1 = {answer[0]}, x2 = {answer[1]}")

print(f" f(x) = {z}")

# Двойственная задача

display\_dual\_problem(obj, constraints, rhs)

# Вектор C\_B (из целевой функции по базису)

C\_B = []

for b in basis:

if b < len(obj):

C\_B.append(Rational(obj[b]))

else:

C\_B.append(Rational(0))

# Вычисление D⁻¹

D\_inv = get\_D\_inverse(constraints, basis, len(obj))

# Проверка первой теоремы двойственности

check\_duality\_theorem(C\_B, D\_inv, answer)

# Проверка второй теоремы двойственности

y\_opt = check\_second\_duality\_theorem(table, basis, len(constraints), len(obj), rhs)

# Проверка третьей теоремы двойственности

y\_opt = check\_third\_duality\_theorem(y\_opt, rhs)

# Анализ влияния на Gmax (без вывода ресурсов)

analyze\_stability\_and\_impact(y\_opt, rhs, z)