

|  |
| --- |
| 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-15-22 Оганнисян Г. А. *(Ф. И.О. студента)*

Преподаватель Железняк Л.М.

*(Ф.И.О. преподавателя)*

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

Москва 2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc166546558)

[1 ДВОЙСТВЕННАЯ ЗАДАЧА 4](#_Toc166546559)

[1.1 Постановка задачи 4](#_Toc166546560)

[1.2 Математическая модель исходной задачи 4](#_Toc166546561)

[1.3 Первая теорема двойственности 5](#_Toc166546562)

[1.4 Вторая теорема двойственности 7](#_Toc166546563)

[1.5 Третья теорема двойственности 9](#_Toc166546564)

[1.6 Результат работы программы 12](#_Toc166546565)

[ЗАКЛЮЧЕНИЕ 13](#_Toc166546566)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 14](#_Toc166546567)

[ПРИЛОЖЕНИЯ 15](#_Toc166546568)

ВВЕДЕНИЕ

Двойственная задача, как дополнительная к прямой задаче оптимизации, играет важную роль в определении нижней границы для значения целевой функции прямой задачи. Процесс ее решения обычно включает следующие этапы:

1. Приведение прямой задачи к стандартной форме для задач линейного программирования.
2. Формулирование уравнений для двойственной задачи на основе коэффициентов ограничений прямой задачи.
3. Решение уравнений двойственной задачи с использованием методов линейного программирования.
4. Определение оптимального значения целевой функции прямой задачи и сравнение его с результатом двойственной задачи.

Двойственная задача находит широкое применение в различных областях, включая экономику, финансы, транспортные системы и производство. Она помогает оценить эффективность ресурсов, их стоимость и оптимальное распределение для достижения оптимального решения задачи.

**1 ДВОЙСТВЕННАЯ ЗАДАЧА**

* 1. **Постановка задачи**

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

***Задача.*** Фабрика может производить тарелки и кружки. На производство тарелки идет 5 единиц материала, на производство кружки – 20 единиц (керамики). Тарелка требует 10 человеко-часа, кружка – 15. На производство тарелки тратится 0,5 кВатт электроэнергии, кружки – 0,3. Расходы при производстве тарелки равны 1 рубль, а кружки – 2 рубля. Имеется 400 единиц материала и 450 человеко-часов, 25 кВатт энергии и объем накладных расходов равен 300 рублей. Эти данные представлены в таблице 1.1.

*Таблица 1.1. Исходные данные задачи.*

|  |  |  |  |
| --- | --- | --- | --- |
| Ресурс | Товар | | Объем  ресурса |
| Тарелки | Кружки |
| Материал | 5 | 20 | 400 |
| Человеко-часы | 10 | 15 | 450 |
| Электроэнергия | 0,5 | 0,3 | 25 |
| Расходы | 1 | 2 | 300 |

Прибыль при производстве тарелки – 20 рублей, при производстве кружки – 50 рублей. Сколько надо сделать тарелок и кружек, чтобы получить максимальную прибыль?

* 1. **Математическая модель исходной задачи**

Пусть х1 – количество тарелок, х2 – количество кружек. Максимальный прибыль составит 20х1 + 50х2.

Ограничения задачи:

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

.

Запишем двойственную задачу. Найти минимум функции.

При ограничениях:

* 1. **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 изделий, 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, 𝑥2. Соответствующие этим переменным векторы , в разложении используются для формирования столбцов матрицы D.

Тогда,

Для вычисления обратной матрицы 𝐷-1 запишем матрицу 𝐷 дописав к ней справа единичную матрицу.

Для нахождения обратной матрицы 𝐷-1 используем элементарные преобразования над строками матрицы. Таким образом, преобразуются левая часть полученной матрицы в единичную.

Делим 1-ую строку на 6;

от 2 строки отнимаем 1 строку, умноженную на 2; от 3 строки отнимаем 1 строку, умноженную на 4;

3-ую строку делим на 12.66;

от 1 строки отнимаем 3 строку, умноженную на 0.833; от 2 строки отнимаем 3 строку, умноженную на 2.33;

Запишем обратную матрицу.

Базисными переменными в симплекс-таблице являются , тогда

При этом минимальное значение целевой функции двойственной задачи

совпадает с максимальным значением 𝑓𝑚𝑎𝑥 = [товаров] прямой задачи, что является результатом взаимодвойственности. Таким образом,

* 1. **Вторая теорема двойственности**

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

Итак, имеем оптимальное решение прямой задачи: за единицу времени объем изделий по продукции I – 𝑥1 = 0.65789; за единицу времени объем изделий по продукции II – 𝑥2 = 9.2105263; максимальный доход от продажи 𝑓𝑚𝑎𝑥 = 282.894 [товаров/единица времени]. 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, для этого умножим 1-ое уравнение на 4 и вычтем из полученного 1-го уравнения 2-ое.

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 1.2 – Выполнение неравенств прямой задачи*

|  |  |  |
| --- | --- | --- |
| Ограничение | Расчет | Вывод |
| 6х1 + 5х3 ≤ 50 | 6\*0.65789 + 5\*9.2105263 = 50  50 = 50 | Первое ограничение прямой задачи выполняется как равенство, остается спрос на изделия по продукции I, полностью используется в оптимальном плане, является дефицитным и его оценка согласно второй теоремы двойственности отлична от нуля (𝑦1 ≠ 0). |
| 2х1 + 4х3 ≤ 128 | 2\*0.65789 + 4\*9.2105263 <128  38<128 | Второе ограничение прямой задачи выполняется как строгое неравенство, значит, этот ресурс не является дефицитным и его оценка в оптимальном плане равна нулю (𝑦2 = 0). |
| 4х1 + 16х3 ≤ 600 | 4\*0.65789 + 16\*9.2105263 = 150  150 = 150 | Третье ограничение прямой задачи выполняется как равенство, остается спрос на изделия по продукции I, полностью используется в оптимальном плане, является дефицитным и его оценка согласно второй теоремы двойственности отлична от нуля (𝑦3 ≠ 0). |
| х1 ≥ 0 | 0.65789 > 0 | Первое ограничение в двойственной задаче будет равенством 6𝑦1 + 2𝑦2 + 4*у3* = 10, т.е. весь его запас полностью используется в оптимальном плане, он является дефицитным |

*Продолжение таблицы 1.2 – Выполнение неравенств прямой задачи*

|  |  |  |
| --- | --- | --- |
| x2 ≥ 0 | 0 = 0 | Второе ограничение в двойственной задаче будет равенством 𝑦2 = 0, т.е. в процессе производства не используется является не дефицитном. |
| x3 ≥ 0 | 9.2105263 > 0 | Третье ограничение в двойственной задаче будет равенством 5𝑦1 + 4𝑦2 + 16*у3* = 30, т.е. весь его запас полностью используется в оптимальном плане, он является дефицитным |

* 1. **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 (Продукция I)*. Найдем нижнюю границу. В первом столбце обратной матрицы один положительный элемент (0.210), ему соответствует индекс базисной переменной оптимального плана (150).

Найдем верхнюю границу. В первом столбце минимальное по модулю значение (), которое соответствует индексу базисной переменной оптимального плана (150).

Таким образом, получаем

Тогда первый ресурс может изменяться в интервале:

*Ресурс 2 (Ограничение по объему производства изделий по продукции I по сравнению с объемом производства изделий по продукции II).* Рассматриваем второй столбец обратной матрицы, в котором найдем нижнюю границу, во втором столбце один положительный элемент (1). Ему соответствуют индекс базисной переменной оптимального плана (150).

Тогда находим нижнюю границу.

Найдем верхнюю границу, во втором столбце минимальное по модулю значение (0), которое соответствует индексу базисной переменной оптимального плана (150).

Получаем

Тогда второй ресурс может изменяться в интервале:

*Ресурс 3 (Продукция II).* Рассматриваем третий столбец обратной матрицы, в котором один положительный элемент (0.078). Данному элементу соответствует индекс соответствующего базисного переменного оптимального плана (150).

Находим нижнюю границу.

Найдем верхнюю границу, в третьем столбце минимальное по модулю значение (), которое соответствует индексу базисной переменной оптимального плана (150).

Получаем

Получаем интервал устойчивости оценок по отношению к третьему ограничению:

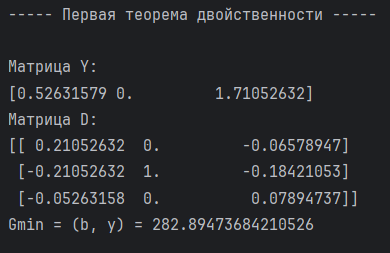
Далее оценим влияние изменения объема ресурсов на величину максимального производства изделий. Как известно, это дефицитные ресурсы Введем верхние границы в формулу:

Совместное влияние изменений этих ресурсов приводит к изменению максимального производства изделий 𝐺𝑚𝑎𝑥 на величину:

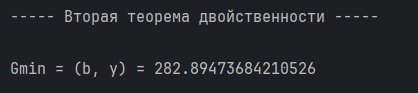
Следовательно, оптимальное значение целевой функции при максимальном изменении ресурсов:

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

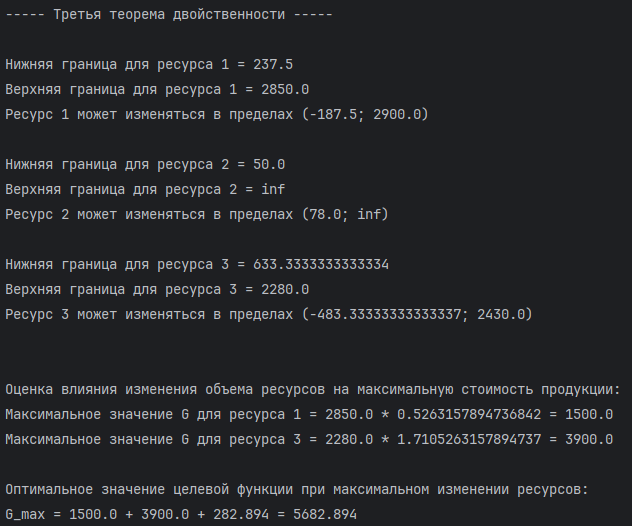
* 1. **Результат работы программы**



**Рисунок 1 – Первая теорема двойственности**



**Рисунок 2 – Вторая теорема двойственности**



**Рисунок 3 – Третья теорема двойственности**

**ЗАКЛЮЧЕНИЕ**

В ходе работы, мы успешно применили теорию двойственности линейного программирования для решения задачи оптимизации. Сначала мы сформулировали прямую задачу в стандартной форме с ограничениями и целевой функцией. Затем мы составили и решили двойственную задачу, которая помогла нам найти оптимальное значение двойственной целевой функции и установить нижнюю границу для значения целевой функции прямой задачи.

В процессе решения мы выявили преимущества данного метода, такие как возможность оценки эффективности использования ресурсов, определение оптимального распределения ресурсов и нахождение нижней границы для значения целевой функции прямой задачи. Однако мы также обратили внимание на некоторые недостатки, включая необходимость дополнительных вычислений, наличие не всегда единственного оптимального решения и важность аккуратной формулировки прямой задачи для корректного вычисления двойственной задачи.

В целом, использование теории двойственности линейного программирования доказало свою эффективность и значимость в решении практических задач оптимизации.

**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.*

|  |
| --- |
| import numpy as np  class VariableValues:  def \_\_init\_\_(self, values, idx):  self.values = values  self.idx = idx  class DeficitResource:  def \_\_init\_\_(self, val, idx):  self.val = val  self.idx = idx  def extract\_c\_vector(table):  return table[0][1:-1]  def transpose\_matrix(mat):  return np.transpose(mat)  def extract\_b\_vector(table):  return [row[-1] for row in table[1:-1]]  def extract\_a\_matrix(table):  return np.array(table[1:-1, 1:-1])  def extract\_cb\_vector(table):  return [row[0] for row in table[1:-1]]  def construct\_d\_matrix(basis):  return np.array([item.values for item in basis])  def invert\_matrix(mat):  return np.linalg.inv(mat)  def dot\_product(mat1, mat2):  return np.dot(mat2, mat1)  def calculate\_gmin(b, y):  return np.dot(b, y)  def first\_duality\_theorem(last\_table, basis, b):  d\_matrix = construct\_d\_matrix(basis)  d\_inv = invert\_matrix(d\_matrix)  cb\_vector = extract\_cb\_vector(last\_table)  y\_vector = dot\_product(d\_inv, cb\_vector)  gmin = calculate\_gmin(b, y\_vector)  return y\_vector, d\_inv, gmin  def find\_max(plan):  half\_len = len(plan) // 2  max\_val = max(plan[:half\_len])  max\_idx = plan.index(max\_val)  return max\_val, max\_idx  def compute\_borders(max\_x, a\_matrix, b\_vector):  y\_vector = []  for i in range(len(b\_vector)): |

*Продолжение листинга А.1. Реализация двойственной задачи.*

|  |
| --- |
| product = a\_matrix[max\_x[1]][i] \* max\_x[0]  if product < b\_vector[i]:  y\_vector.append(0.0)  elif product == b\_vector[i]:  y\_vector.append(4.0)  return y\_vector  def detect\_deficit(y\_vector):  for i, val in enumerate(y\_vector):  if val != 0.0:  return DeficitResource(val, i)  print("Дефицитный ресурс не обнаружен!")  return None  def second\_duality\_theorem(last\_table, a\_matrix, b\_vector, plan):  max\_x = find\_max(plan)  d\_matrix = construct\_d\_matrix(basis)  d\_inv = invert\_matrix(d\_matrix)  cb\_vector = extract\_cb\_vector(last\_table)  y\_vector = dot\_product(d\_inv, cb\_vector)  gmin = calculate\_gmin(b\_vector, y\_vector)  return detect\_deficit(y\_vector), gmin  def find\_lower\_bound(d\_matrix, na0, col):  values = [min(na0) / abs(d\_matrix[i][col]) for i in range(len(d\_matrix[0])) if d\_matrix[i][col] > 0]  return min(values) if values else np.inf  def find\_upper\_bound(d\_matrix, na0, col):  values = [max(na0) / abs(d\_matrix[i][col]) for i in range(len(d\_matrix[0])) if d\_matrix[i][col] < 0]  return max(values) if values else np.inf  def calculate\_resource\_range(lower\_bound, upper\_bound, col, b\_vector):  return b\_vector[col] - lower\_bound, upper\_bound + b\_vector[col]  def third\_duality\_theorem(d\_matrix, b\_vector, a0, y\_vector, last\_table, deficit\_resource, na0):  print("\n----- Третья теорема двойственности -----\n")  lower\_bounds = []  upper\_bounds = []  for i in range(len(d\_matrix)):  lower = find\_lower\_bound(d\_matrix, na0, i)  upper = find\_upper\_bound(d\_matrix, na0, i)  lower\_bounds.append(lower)  upper\_bounds.append(upper)  print(f"Нижняя граница для ресурса {i + 1} = {lower}")  print(f"Верхняя граница для ресурса {i + 1} = {upper}")  x\_range = b\_vector[i] - lower  y\_range = upper + b\_vector[i]  print(f"Ресурс {i + 1} может изменяться в пределах ({x\_range}; {y\_range})\n")  print("\nОценка влияния изменения объема ресурсов на максимальную стоимость продукции:")  g\_max\_1 = upper\_bounds[0] \* y\_vector[0]  print(f"Максимальное значение G для ресурса {deficit\_resource.idx + 1} = {upper\_bounds[0]} \* {y\_vector[0]} = {g\_max\_1}")  g\_max\_3 = upper\_bounds[2] \* y\_vector[2]  print(f"Максимальное значение G для ресурса {deficit\_resource.idx + 3} = {upper\_bounds[2]} \* {y\_vector[2]} = {g\_max\_3}") |

*Продолжение листинга А.1. Реализация двойственной задачи.*

|  |
| --- |
| print("\nОптимальное значение целевой функции при максимальном изменении ресурсов:")  g\_max = g\_max\_1 + g\_max\_3 + last\_table[-1][-1]  print(f"G\_max = {g\_max\_1} + {g\_max\_3} + {last\_table[-1][-1]} = {g\_max}")  def print\_simplex\_table(table):  for row in table:  print(" ".join(f"{cell:10.2f}" for cell in row))  print("\n")  if \_\_name\_\_ == "\_\_main\_\_":  f = "50\*x1 + 128\*x2 + 150\*x3"  initial\_table = np.array([  [float('inf'), 10.0, 30.0, float('inf')],  [0.0, 6.0, 5.0, 50.0],  [0.0, 2.0, 4.0, 128.0],  [0.0, 4.0, 16.0, 150.0],  [0.0, -10.0, -30.0, 0.0]  ])  last\_table = np.array([  [float('inf'), 0, 0, float('inf')],  [10.0, 0.210, -0.065, 0.65789473684210526315789473684211],  [0, -0.210, -0.184, 89.8],  [30.0, -0.052, 0.0788, 9.2105263157894736842105263157895],  [0, 0.5263, 1.70, 282.894]  ])  basis = [  VariableValues([6.0, 0.0, 5.0], 1),  VariableValues([2.0, 1.0, 4.0], 4),  VariableValues([4.0, 0.0, 16.0], 2),  ]  plan = [0.0, 0.0, 10, 89.8, 30]  a0 = [0.65789473684210526315789473684211, 0.0, 9.2105263157894736842105263157895]  na0 = [50.0, 128.0, 150.0]  c\_vector = extract\_c\_vector(initial\_table)  print("Вектор C:")  print(c\_vector)  b\_vector = extract\_b\_vector(initial\_table)  print("Вектор B:")  print(b\_vector)  a\_matrix = extract\_a\_matrix(initial\_table)  print("Матрица A:")  print(a\_matrix)  print("Транспонированная матрица A:")  a\_matrix = transpose\_matrix(a\_matrix)  print(a\_matrix)  print("Двойственная задача:")  print(f"g(y) = (b, y) = {f.replace('x', 'y')} -> min")  print("Ограничения:")  y\_vector, d\_matrix\_inv, gmin = first\_duality\_theorem(last\_table, basis, b\_vector) |

*Продолжение листинга А.1. Реализация двойственной задачи.*

|  |
| --- |
| print("\n----- Первая теорема двойственности -----\n")  print("Матрица Y:")  print(y\_vector)  print("Матрица D:")  print(d\_matrix\_inv)  print(f"Gmin = (b, y) = {gmin}\n")  print("\n----- Вторая теорема двойственности -----\n")  deficit\_resource, gmin = second\_duality\_theorem(last\_table, a\_matrix, b\_vector, plan)  print(f"Gmin = (b, y) = {gmin}\n")  third\_duality\_theorem(d\_matrix\_inv, b\_vector, a0, y\_vector, last\_table, deficit\_resource, na0) |