БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра информатики

Факультет КСИС

Специальность ИиТП

Контрольная работа №1: Линейное программирование

по дисциплине «Методы оптимизации и управления»

Выполнил студент: Драгун О.В.

группа 893551

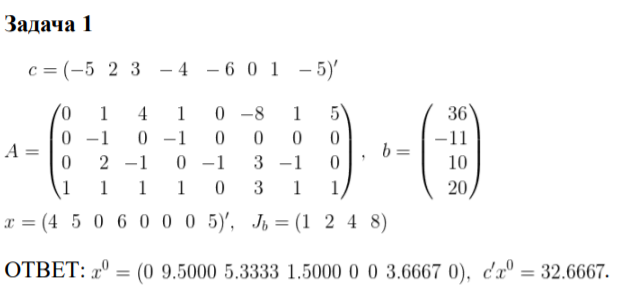
Зачетная книжка № 2520050

Руководитель: Алёхина Алина Энодиевна

Минск 2021

# Симплекс-метод

Решить симплекс-методом задачу линейного программирования вида) с заданными исходными данными и начальным базисным планом.



### **Вывод программы, не требующий дополнительных комментариев:**

Матрицы:

Ab =

[[0. 0. 0. 0.]

[0. 0. 0. 0.]

[0. 0. 0. 0.]

[0. 0. 0. 0.]]

Ab\_inv=

[[-0.2 0.8 0. 1. ]

[ 0. 0. 0.5 0. ]

[-0. -1. -0.5 -0. ]

[ 0.2 0.2 0. 0. ]]

Итерация 1

cb = [-5, 2, -4, -5]

Вектр потенциалов u = [ 0. -1. 3. -5.]

uA [-5. 2. -8. -4. -3. -6. -8. -5.]

delta (вектор оценок) [0.0, 0.0, -11.0, 0.0, 3.0, -6.0, -9.0, 0.0]

Выберем индекс j0: 2

Построим вектор z [ 0.2 -0.5 0.5 0.8]

Tetta:

[20.000000000000004, inf, 12.0, 6.25]

Tetta0:

6.25

Tetta index:

3

Новый план: [2.75, 8.125, 6.25, 2.875, 0, 0, 0, 0]

[ 0.2 -0.5 0.5 0.8]

True

Матрица обратная базисной

[[-0.25 0.75 0. 1. ]

[ 0.125 0.125 0.5 0. ]

[-0.125 -1.125 -0.5 0. ]

[ 0.25 0.25 0. 0. ]]

[[ 0. 1. 1. 4.]

[ 0. -1. -1. 0.]

[ 0. 2. 0. -1.]

[ 1. 1. 1. 1.]]

Итерация 2

cb = [-5, 2, -4, 3]

Вектр потенциалов u = [ 2.75 1.75 3. -5. ]

uA [ -5. 2. 3. -4. -3. -28. -5.25 8.75]

delta (вектор оценок) [0.0, 0.0, 0.0, 0.0, 3.0, -28.0, -6.25, 13.75]

Выберем индекс j0: 5

Построим вектор z [ 5. 0.5 -0.5 -2. ]

Tetta:

[0.55, 16.25, inf, inf]

Tetta0:

0.55

Tetta index:

0

Новый план: [0, 7.85, 7.35, 3.15, 0, 0.55, 0, 0]

[ 5. 0.5 -0.5 -2. ]

True

Матрица обратная базисной

[[-0.05 0.15 0. 0.2 ]

[ 0.15 0.05 0.5 -0.1 ]

[-0.15 -1.05 -0.5 0.1 ]

[ 0.15 0.55 0. 0.4 ]]

[[-8. 1. 1. 4.]

[ 0. -1. -1. 0.]

[ 3. 2. 0. -1.]

[ 3. 1. 1. 1.]]

Итерация 3

cb = [0, 2, -4, 3]

Вектр потенциалов u = [1.35 5.95 3. 0.6 ]

uA [ 6.0000000e-01 2.0000000e+00 3.0000000e+00 -4.0000000e+00

-3.0000000e+00 -8.8817842e-16 -1.0500000e+00 7.3500000e+00]

delta (вектор оценок) [5.6, -8.881784197001252e-16, 4.440892098500626e-16, 0.0, 3.0, -8.881784197001252e-16, -2.05, 12.35]

Выберем индекс j0: 6

Построим вектор z [ 0.15 -0.45 0.45 0.55]

Tetta:

[3.6666666666666665, inf, 7.000000000000001, 13.363636363636362]

Tetta0:

3.6666666666666665

Tetta index:

0

Новый план: [0, 9.5, 5.333333333333333, 1.5000000000000002, 0, 0, 3.6666666666666665, 0]

[ 0.15 -0.45 0.45 0.55]

True

Матрица обратная базисной

[[-0.33333333 1. 0. 1.33333333]

[ 0. 0.5 0.5 0.5 ]

[ 0. -1.5 -0.5 -0.5 ]

[ 0.33333333 0. 0. -0.33333333]]

[[ 1. 1. 1. 4.]

[ 0. -1. -1. 0.]

[-1. 2. 0. -1.]

[ 1. 1. 1. 1.]]

Итерация 4

cb = [1, 2, -4, 3]

Вектр потенциалов u = [0.66666667 8. 3. 3.33333333]

uA [ 3.33333333 2. 3. -4. -3. 13.66666667

1. 6.66666667]

delta (вектор оценок) [8.333333333333332, 0.0, 0.0, 0.0, 3.0, 13.666666666666666, 0.0, 11.666666666666668]

Оптимальный план x: [0, 9.5, 5.333333333333333, 1.5000000000000002, 0, 0, 3.6666666666666665, 0]

Базисные индексы: [6, 1, 3, 2]

Process finished with exit code 0

## Задача протестирована на вариантах 1 и 3 и с ответом совпадает

# Метод потенциалов для решения матричной транспортной задачи

## Задача

Изображение выглядит как стол

Автоматически созданное описание

## Вывод программы:

a [53, 20, 45, 38]

b [15, 31, 10, 3, 18, 79]

plan [[15 31 7 0 0 0]

[ 0 0 3 3 14 0]

[ 0 0 0 0 4 41]

[ 0 0 0 0 0 38]]

j\_b [(0, 1), (2, 4), (1, 2), (0, 0), (1, 4), (0, 2), (2, 5), (1, 3), (3, 5)]

iter 1

plan [[15 31 7 0 0 0]

[ 0 0 3 3 14 0]

[ 0 0 0 0 4 41]

[ 0 0 0 0 0 38]]

j\_b [(0, 1), (2, 4), (1, 2), (0, 0), (1, 4), (0, 2), (2, 5), (1, 3), (3, 5)]

u [-9.0, -2.0, 0.0, 0.0]

v [12.0, 9.0, 12.0, 7.0, 9.0, 0]

delta [[ 0. 0. 0. 3. 6. 9.]

[ -8. -3. 0. 0. 0. 2.]

[-14. -4. -9. -5. 0. 0.]

[-11. -6. -7. -6. 0. 0.]]

imin 2

jmin 0

[[1. 1. 1. 0. 0. 0.]

[0. 0. 1. 1. 1. 0.]

[1. 0. 0. 0. 1. 1.]

[0. 0. 0. 0. 0. 0.]]

[[1. 0. 1. 0. 0. 0.]

[0. 0. 1. 1. 1. 0.]

[1. 0. 0. 0. 1. 1.]

[0. 0. 0. 0. 0. 0.]]

[[1. 0. 1. 0. 0. 0.]

[0. 0. 1. 0. 1. 0.]

[1. 0. 0. 0. 1. 1.]

[0. 0. 0. 0. 0. 0.]]

[[1. 0. 1. 0. 0. 0.]

[0. 0. 1. 0. 1. 0.]

[1. 0. 0. 0. 1. 0.]

[0. 0. 0. 0. 0. 0.]]

cycle [[1. 0. 1. 0. 0. 0.]

[0. 0. 1. 0. 1. 0.]

[1. 0. 0. 0. 1. 0.]

[0. 0. 0. 0. 0. 0.]]

min = 3 (min\_i, min\_j) = (1, 2)

[[-3. 0. 3. 0. 0. 0.]

[ 0. 0. -3. 0. 3. 0.]

[ 3. 0. 0. 0. -3. 0.]

[ 0. 0. 0. 0. 0. 0.]]

--------------------

iter 2

plan [[12. 31. 10. 0. 0. 0.]

[ 0. 0. 0. 3. 17. 0.]

[ 3. 0. 0. 0. 1. 41.]

[ 0. 0. 0. 0. 0. 38.]]

j\_b [(0, 1), (2, 4), (0, 0), (1, 4), (0, 2), (2, 5), (1, 3), (3, 5), (2, 0)]

u [5.0, -2.0, 0.0, 0.0]

v [-2.0, -5.0, -2.0, 7.0, 9.0, 0]

delta [[ 0. 0. 0. -11. -8. -5.]

[ 6. 11. 14. 0. 0. 2.]

[ 0. 10. 5. -5. 0. 0.]

[ 3. 8. 7. -6. 0. 0.]]

imin 0

jmin 3

[[1. 1. 1. 1. 0. 0.]

[0. 0. 0. 1. 1. 0.]

[1. 0. 0. 0. 1. 1.]

[0. 0. 0. 0. 0. 0.]]

[[1. 0. 1. 1. 0. 0.]

[0. 0. 0. 1. 1. 0.]

[1. 0. 0. 0. 1. 1.]

[0. 0. 0. 0. 0. 0.]]

[[1. 0. 0. 1. 0. 0.]

[0. 0. 0. 1. 1. 0.]

[1. 0. 0. 0. 1. 1.]

[0. 0. 0. 0. 0. 0.]]

[[1. 0. 0. 1. 0. 0.]

[0. 0. 0. 1. 1. 0.]

[1. 0. 0. 0. 1. 0.]

[0. 0. 0. 0. 0. 0.]]

cycle [[1. 0. 0. 1. 0. 0.]

[0. 0. 0. 1. 1. 0.]

[1. 0. 0. 0. 1. 0.]

[0. 0. 0. 0. 0. 0.]]

min = 1.0 (min\_i, min\_j) = (2, 4)

[[-1. 0. 0. 1. 0. 0.]

[ 0. 0. 0. -1. 1. 0.]

[ 1. 0. 0. 0. -1. 0.]

[ 0. 0. 0. 0. 0. 0.]]

--------------------

iter 3

plan [[11. 31. 10. 1. 0. 0.]

[ 0. 0. 0. 2. 18. 0.]

[ 4. 0. 0. 0. 0. 41.]

[ 0. 0. 0. 0. 0. 38.]]

j\_b [(0, 1), (0, 0), (1, 4), (0, 2), (2, 5), (1, 3), (3, 5), (2, 0), (0, 3)]

u [5.0, 9.0, 0.0, 0.0]

v [-2.0, -5.0, -2.0, -4.0, -2.0, 0]

delta [[ 0. 0. 0. 0. 3. -5.]

[-5. 0. 3. 0. 0. -9.]

[ 0. 10. 5. 6. 11. 0.]

[ 3. 8. 7. 5. 11. 0.]]

imin 1

jmin 5

[[1. 1. 1. 1. 0. 0.]

[0. 0. 0. 1. 1. 1.]

[1. 0. 0. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

[[1. 0. 1. 1. 0. 0.]

[0. 0. 0. 1. 1. 1.]

[1. 0. 0. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

[[1. 0. 0. 1. 0. 0.]

[0. 0. 0. 1. 1. 1.]

[1. 0. 0. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

[[1. 0. 0. 1. 0. 0.]

[0. 0. 0. 1. 0. 1.]

[1. 0. 0. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

cycle [[1. 0. 0. 1. 0. 0.]

[0. 0. 0. 1. 0. 1.]

[1. 0. 0. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

min = 2.0 (min\_i, min\_j) = (1, 3)

[[-2. 0. 0. 2. 0. 0.]

[ 0. 0. 0. -2. 0. 2.]

[ 2. 0. 0. 0. 0. -2.]

[ 0. 0. 0. 0. 0. 0.]]

--------------------

iter 4

plan [[ 9. 31. 10. 3. 0. 0.]

[ 0. 0. 0. 0. 18. 2.]

[ 6. 0. 0. 0. 0. 39.]

[ 0. 0. 0. 0. 0. 38.]]

j\_b [(0, 1), (0, 0), (1, 4), (0, 2), (2, 5), (3, 5), (2, 0), (0, 3), (1, 5)]

u [5.0, 0.0, 0.0, 0.0]

v [-2.0, -5.0, -2.0, -4.0, 7.0, 0]

delta [[ 0. 0. 0. 0. -6. -5.]

[ 4. 9. 12. 9. 0. 0.]

[ 0. 10. 5. 6. 2. 0.]

[ 3. 8. 7. 5. 2. 0.]]

imin 0

jmin 4

[[1. 1. 1. 1. 1. 0.]

[0. 0. 0. 0. 1. 1.]

[1. 0. 0. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

[[1. 0. 1. 1. 1. 0.]

[0. 0. 0. 0. 1. 1.]

[1. 0. 0. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

[[1. 0. 0. 1. 1. 0.]

[0. 0. 0. 0. 1. 1.]

[1. 0. 0. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

[[1. 0. 0. 0. 1. 0.]

[0. 0. 0. 0. 1. 1.]

[1. 0. 0. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

cycle [[1. 0. 0. 0. 1. 0.]

[0. 0. 0. 0. 1. 1.]

[1. 0. 0. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

min = 9.0 (min\_i, min\_j) = (0, 0)

[[-9. 0. 0. 0. 9. 0.]

[ 0. 0. 0. 0. -9. 9.]

[ 9. 0. 0. 0. 0. -9.]

[ 0. 0. 0. 0. 0. 0.]]

--------------------

iter 5

plan [[ 0. 31. 10. 3. 9. 0.]

[ 0. 0. 0. 0. 9. 11.]

[15. 0. 0. 0. 0. 30.]

[ 0. 0. 0. 0. 0. 38.]]

j\_b [(0, 1), (1, 4), (0, 2), (2, 5), (3, 5), (2, 0), (0, 3), (1, 5), (0, 4)]

u [-1.0, 0.0, 0.0, 0.0]

v [-2.0, 1.0, 4.0, 2.0, 7.0, 0]

delta [[ 6. 0. 0. 0. 0. 1.]

[ 4. 3. 6. 3. 0. 0.]

[ 0. 4. -1. 0. 2. 0.]

[ 3. 2. 1. -1. 2. 0.]]

imin 2

jmin 2

[[0. 1. 1. 1. 1. 0.]

[0. 0. 0. 0. 1. 1.]

[1. 0. 1. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

[[0. 1. 1. 1. 1. 0.]

[0. 0. 0. 0. 1. 1.]

[0. 0. 1. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

[[0. 0. 1. 1. 1. 0.]

[0. 0. 0. 0. 1. 1.]

[0. 0. 1. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

[[0. 0. 1. 0. 1. 0.]

[0. 0. 0. 0. 1. 1.]

[0. 0. 1. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

cycle [[0. 0. 1. 0. 1. 0.]

[0. 0. 0. 0. 1. 1.]

[0. 0. 1. 0. 0. 1.]

[0. 0. 0. 0. 0. 0.]]

min = 9.0 (min\_i, min\_j) = (1, 4)

[[ 0. 0. -9. 0. 9. 0.]

[ 0. 0. 0. 0. -9. 9.]

[ 0. 0. 9. 0. 0. -9.]

[ 0. 0. 0. 0. 0. 0.]]

--------------------

iter 6

plan [[ 0. 31. 1. 3. 18. 0.]

[ 0. 0. 0. 0. 0. 20.]

[15. 0. 9. 0. 0. 21.]

[ 0. 0. 0. 0. 0. 38.]]

j\_b [(0, 1), (0, 2), (2, 5), (3, 5), (2, 0), (0, 3), (1, 5), (0, 4), (2, 2)]

u [0.0, 0.0, 0.0, 0.0]

v [-2.0, 0.0, 3.0, 1.0, 6.0, 0]

delta [[5. 0. 0. 0. 0. 0.]

[4. 4. 7. 4. 1. 0.]

[0. 5. 0. 1. 3. 0.]

[3. 3. 2. 0. 3. 0.]]

Оптимальный план:

[[ 0. 31. 1. 3. 18. 0.]

[ 0. 0. 0. 0. 0. 20.]

[15. 0. 9. 0. 0. 21.]

[ 0. 0. 0. 0. 0. 38.]]

j\_b [(0, 1), (0, 2), (2, 5), (3, 5), (2, 0), (0, 3), (1, 5), (0, 4), (2, 2)]

Общие затраты на перевозку всей продукции для оптимального плана составляют: 111.0

Process finished with exit code 0

## Ответ из иллюстраций к теме 1

## Изображение выглядит как стол Автоматически созданное описание Программа протестирована на 3 примерах и работает безошибочно

# Вывод

Научился решать задачи линейного программирования двойственным симплекс методом, изучил метод потенциалов для решения транспортной задачи, а так же написал необходимый код и протестировал его на достаточном кол-ве примеров, чем автоматизировал процесс решения подобных задач в будущем.