МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №1
по курсу «Математическая экономика»

Выполнил: А. О. Тояков
Группа: М8О-407Б-18
Преподаватель: В. М. Подгорная
Дата:
Оценка:

Москва, 2022 г.

ПОСТАНОВКА ЗАДАЧИ

Рассматривается задача планирования производства

$$c^{\top}x \to \max_{x \in X},$$

где

$$X = \{u \in \mathbb{R}^n \mid Ax \leqslant b,\ x \geqslant 0\},$$

$c \in \mathbb{R}^n$ — вектор цен на продукцию, $b \in \mathbb{R}^m$ — вектор ограничений на ресурсы, $A \in \mathbb{R}^{m \times n}$ — технологическая матрица, x — план производства.

$n = 15, m = 25, b_i = 100 + i, c_i = 70 - i, a_{ij} = 1 + ((j + 23)i + j^2 + i^3 + 3(i + 7)) \mod 34$

РЕШЕНИЕ

Для решения мы будем использовать язык программирования Python и пакет для решения задач линейного программирования pulp.

Моделируем данные в соответствии с вариантом:

```
n = 15
m = 25

b = np.array([[100 + i] for i in range(m)])
c = np.array([[70 - i] for i in range(n)])

A = []

for i in range(m):
    l = []
    for j in range(n):
        l.append(1 + ((j + 23) * i + j ** 2 + i ** 3 + 3 * (i + 7)) % 34)
    A.append(l)


A = np.array(A)
```

Ставим прямую задачу линейного программирования (ЗЛП):

```
model_primal = LpProblem(name="resource-allocation", sense=LpMaximize)

x = [LpVariable(name=f"x{i}", lowBound=0) for i in range(1, n+1)]

for i in range(m):
    model_primal += np.dot(A[i], x) <= b[i,0]

model_primal += lpSum(np.dot(c.T, x))
```

Решаем ЗЛП и получаем решение:

```python
status_primal = model_primal.solve()

print(f"status: {model_primal.status}, {LpStatus[model_primal.status]}")

print(f"objective: {model_primal.objective.value()}")

for var in model_primal.variables():
    print(f"{var.name}: {var.value()}")

for name, constraint in model_primal.constraints.items():
    print(f"{name}: {constraint.value()}")
```

```
status: 1, Optimal
objective: 395.722650185
x1: 0.57891566
x10: 0.83144578
x11: 0.0
x12: 0.62626506
x13: 0.42361446
x14: 0.031566265
x15: 0.0
x2: 0.48674699
x3: 0.81566265
x4: 0.21843373
x5: 0.81313253
x6: 0.45518072
x7: 0.84722892
x8: 0.0
x9: 0.0
_C1: -5.499999633507002e-08
_C2: -25.04939768499999
_C3: -9.4999990152278e-08
_C4: -1.9500002013117523e-07
_C5: -8.500000636679772e-08
_C6: -23.976144854999998
_C7: 2.4999993186725078e-08
_C8: -1.749999952727066e-07
_C9: -3.2197592549999943
_C10: -5.499999156111102e-08
_C11: -2.7500000943092573e-07
_C12: -1.3500000528576805e-07
_C13: -28.26915676499999
_C14: -34.62265069499999
_C15: -26.573253094999995
_C16: -1.8500000398269378e-07
_C17: -17.000000085
_C18: -48.48891577499999
_C19: -39.45228932499999
_C20: -14.402891734999985
_C21: -5.19421691500001
_C22: -32.92674690500001
_C23: -9.573253215
_C24: -48.402891615000016
_C25: -1.7499999260817134e-07
```

Сформулируем двойственную задачу и получим её решение:

$$(b, y) \to \min_{y \in Y}$$
$$Y = \{y | A^T y \geq c, y \geq 0\}$$

```
status: 1, Optimal
objective: 395.72264984
y1: 0.27506024
y10: 0.59903614
y11: 0.13325301
y12: 0.4413253
y13: 0.0
y14: 0.0
y15: 0.0
y16: 0.26650602
y17: 0.0
y18: 0.0
y19: 0.0
y2: 0.0
y20: 0.0
y21: 0.0
y22: 0.0
y23: 0.0
y24: 0.0
y25: 0.22493976
y3: 0.47433735
y4: 0.39120482
y5: 0.34963855
y6: 0.0
y7: 0.29096386
y8: 0.23228916
y9: 0.0
_C1: -2.199999951102427e-07
_C2: -1.099999948905861e-07
_C3: -2.8000000429351246e-07
_C4: -4.999999525523435e-08
_C5: -9.99999993922529e-08
_C6: -9.000000122938445e-08
_C7: -1.9999994549380062e-08
_C8: 8.770361309999997
_C9: 20.367228720000007
_C10: -1.2999998943996616e-07
_C11: 11.056144319999998
_C12: -1.300000000981072e-07
_C13: -3.999999620418748e-08
_C14: 1.0999999933147819e-07
_C15: 23.484578239999998
```

Согласно слабой теореме двойственности (c, x) ≤ (b, y) оптимальное значение целевой функции, получаемое при решении прямой задачи, всегда будет меньше или равно значению, получаемому при решении двойственной задачи. А поскольку наши решения совпадают, то решение ЗЛП является оптимальным, согласно сильной теореме двойственности. Максимальной двойственной переменной является y10, увеличим её запас на единицу и найдём оптимальное значение целевой функции:

```
b_max = b.copy()
b_max[9, 0] += 1

model_max = LpProblem(name="resource-allocation", sense=LpMaximize)

for i in range(m):
    model_max += np.dot(A[i], x) <= b_max[i,0]

model_max += lpSum(np.dot(c.T, x))

status_max = model_max.solve()
print(f"status: {model_max.status}, {LpStatus[model_max.status]}")
print(f"objective: {model_max.objective.value()}")
```

status: 1, Optimal

objective: 396.32168680599995

Как видно, решение увеличилось. Это связано с тем, что решение y∗ – это теневые цены на ресурсы.

$$f(b) = J(x^*) = (c, x^*) = (b, y^*)$$

$$\nabla f = y^*$$

Из этого следует, что максимального увеличения дохода можно добиться при увеличении запаса ресурса максимальной двойственной переменной.

Изменением другой двойственной переменной проверим то, увеличилась ли функция меньше, чем в случае с изменением максимальной двойственной переменной:

```
b_other = b.copy()
b_other[1] += 1

model_other = LpProblem(name="resource-allocation", sense=LpMaximize)

for i in range(m):
    model_other += np.dot(A[i], x) <= b_other[i,0]

model_other += lpSum(np.dot(c.T, x))

status_other = model_other.solve()
print(f"status: {model_other.status}, {LpStatus[model_other.status]}")
print(f"objective: {model_other.objective.value()}")
```

status: 1, Optimal

objective: 395.722650185

Видно, что полученное нами значение больше, чем изначальное оптимальное значение целевой функции и меньше, чем значение целевой функции при увеличении максимальной двойственной переменной. Найдём решение, считая, что план производства - целочисленный:

```
status: 1, Optimal
objective: 332.0
x1: 1.0
x10: 1.0
x11: 0.0
x12: 0.0
x13: 0.0
x14: 0.0
x15: 0.0
x2: 0.0
x3: 1.0
x4: 1.0
x5: 1.0
x6: 0.0
x7: 0.0
x8: 0.0
x9: 0.0
_C1: -16.0
_C2: -34.0
_C3: -22.0
_C4: -18.0
_C5: -26.0
_C6: -16.0
_C7: -26.0
_C8: -26.0
_C9: -20.0
_C10: -12.0
_C11: -6.0
_C12: -6.0
_C13: -16.0
_C14: -74.0
_C15: -48.0
_C16: -10.0
_C17: -32.0
_C18: -50.0
_C19: -34.0
_C20: -22.0
_C21: -52.0
_C22: -94.0
_C23: -16.0
_C24: -60.0
_C25: -26.0
```

Найдём решение, считая, что объёмы производства только первых [n/2] товаров должны быть целочисленными:

```
status: 1, Optimal
objective: 372.35442992000003
x1: 0.0
x10: 0.59915612
x11: 0.5021097
x12: 0.70042194
x13: 0.69620253
x14: 0.4556962
x15: 0.0
x2: 0.0
x3: 1.0
x4: 0.0
x5: 1.0
x6: 0.0
x7: 1.0
x8: 0.0
x9: 0.0
_C1: -1.9999999523179213e-07
_C2: -21.949367270000003
_C3: -1.8000000334694732e-07
_C4: -23.670886190000004
_C5: -11.90717314
_C6: -37.44303803
_C7: -1.4000000625458142e-07
_C8: -1.300000000981072e-07
_C9: -10.185654160000015
_C10: -1.0999999755512135e-07
_C11: -10.185654139999999
_C12: -9.000000122938445e-08
_C13: -35.72151902
_C14: -40.74261623
_C15: -34.00000006
_C16: -8.607595089999995
_C17: -18.64978918
_C18: -54.37130810999999
_C19: -52.649789160000005
_C20: -30.70042207000001
_C21: -20.371308080000013
_C22: -42.60759503
_C23: -34.143459920000005
_C24: -37.29957825000001
_C25: -3.299578239999999
```

ВЫВОД

Выполнив лабораторную работу № 1, я познакомился с пакетом pulp, который позволяет решать задачи линейного программирования. Мной были решены прямая и двойственная задача, найдены оптимальные значения целевых функций, а также я выявил ресурс, который позволяет максимально увеличить значение целевой функции. Путём сравнения значений были проверены слабая и сильная теоремы двойственности.