МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра ИС

ДОМАШНЯЯ РАБОТА

по дисциплине «Теория принятия решений»

Вариант: 42 (434)

Студент гр. 0372 Самойлов Д.В.

Преподаватель Степулёнко Д. О

Санкт-Петербург

2023

**Условие задачи 1:**

На строительство магистрали периодически доставляются материалы, которые со станции железной дороги вначале поступают на 3 промежуточных склада, а затем непосредственно к 5 объектам магистрали. Два первых склада А1, А2 имеют ограниченную емкость 70 т, и поступившие на них грузы расходуются полностью. Cклад А3 с неограниченной емкостью допускает резервирование грузов от периода к периоду. Транспортные расходы (в условных денежных единицах — ДЕ) на перевозку одной тонны груза со станции на промежуточные склады составляют 2, 6, 3 ДЕ, а со складов к объектам В1–В5 представлены в табл. 1; там же приведены потребности объектов (в тоннах) в течение двух периодов.



Рассматривается работа системы в течение двух периодов при условии доставки на станцию по 150 т груза в каждый из периодов.

Требуется определить план перевозок, обеспечивающий минимум стоимости доставки грузов для двух периодов. Провести анализ чувствительности плана к изменению цен перевозок из первого промежуточного склада.

**Алгоритм решения задачи**:

Объявляем все необходимые переменные и данные:

v - матрица стоимостей доставки груза с промежуточных складов на объекты магистрали.

c - список стоимостей доставки груза со станции до промежуточных складов.

s - список емкостей промежуточных складов.

d - матрица потребностей объектов магистрали в течение двух периодов.

xxx1 - трехмерный массив для хранения переменных x (количество товара, поставленного от складов до магазинов в заданный период).

yyy1 - двумерный массив для хранения переменных y (количество товара, поставленного от станции до складов в заданный период).

Определяем функцию obf для вычисления целевой функции и проведения анализа чувствительности, которая принимает переменные xxx, yyy, vvv, ccc и vvvv в качестве аргументов.

Функция вычисляет и возвращает значение целевой функции, учитывая стоимости доставки и количество товара на складах.

Создаем экземпляр класса LpProblem с указанием типа задачи (минимизация) и задаем имя задачи.

Объявляем переменные, используя функцию LpVariable.

varp1 - переменные x (количество товара, поставленного от складов до магазинов в заданный период).

varp2 - переменные y (количество товара, поставленного от станции до складов в заданный период).

varp3 - бинарные переменные m для учета полного расходования товара на первых двух складах.

Целевая функция определяется с помощью метода LpProblem +=. Создаем строку of и добавляем в нее выражения для всех переменных x и y, умноженные на соответствующие стоимости доставки. Затем добавляем в строку of выражения для переменных y и их стоимостей доставки. Наконец, добавляем в строку of выражения для бинарных переменных m и их стоимостей доставки.

Добавляем целевую функцию в экземпляр задачи с помощью метода LpProblem +=.

Также добавляются ограничения для переменных x с использованием метода LpProblem +=. Устанавливаем ограничение на суммарное количество товара на каждом складе, которое не должно превышать его емкость. Добавляем ограничения для переменных y с использованием метода LpProblem +=. Устанавливаем ограничение на суммарное количество товара, поставленного от станции на каждый склад, которое не должно превышать его потребность. Также добавляем ограничения для бинарных переменных m с использованием метода LpProblem +=, и на количество товара, которое может быть полностью использовано на первых двух складах. Добавляем ограничение на неотрицательность всех переменных.

Вызываем метод solve() для решения задачи и получаем статус решения с помощью атрибута status экземпляра задачи. Если решение найдено (статус равен LpStatusOptimal), то выводим оптимальное значение целевой функции и значения переменных x, y и m.

Для проведения анализа чувствительности вызываем функцию obf:

Создаем копии переменных xxx1, yyy1, vvv1, ccc1 и vvvv1, изменяем стоимости доставки в матрице vvv1 (например, увеличиваем на 10%), вызываем функцию obf с измененными данными и сохраняем новое значение целевой функции. Новое значение сравниваем с оптимальным и выводим информацию о чувствительности плана к изменению цен перевозок.

**Листинг кода:**  
from pulp import \*

v=[[7,9,14,5,1],

[3,8,12,7,6],

[3,6,9,1,13]]

c=[2,6,3] # стоимость доставки от станции до складов

s=[70,70,150] # емкость складов

d=[[26,12,9,66,1],

[65,16,55,9,10]]

xxx1=[[[0 for \_ in range(2)] for \_ in range(5)] for \_ in range(3)]

yyy1=[[0 for \_ in range(2)] for \_ in range(3)]

# целевая функция для анализа чувствительности

def obf(xxx,yyy,vvv,ccc, vvvv):

for j in range(5):

vvv[0][j]=vvvv[j]

z1=sum(xxx[i][j][t]\*vvv[i][j] for i in range(3) for j in range(5) for t in range(2))+sum(yyy[i][t]\*ccc[i] for i in range(3) for t in range(2))

return z1

# Создаем экземпляр класса LpProblem с определенным типом решаемой задачи (minimize)

prob = LpProblem("TransportationProblem", LpMinimize)

# индексы количество товара поставленного от складов до магазинов в заданный период

xijt=[(1,1,1),(1,2,1),(1,3,1),(1,4,1),(1,5,1),

#второй склад

(2,1,1),(2,2,1),(2,3,1),(2,4,1),(2,5,1),

#третий склад

(3,1,1),(3,2,1),(3,3,1),(3,4,1),(3,5,1),

# второй период

(1,1,2),(1,2,2),(1,3,2),(1,4,2),(1,5,2),

(2,1,2),(2,2,2),(2,3,2),(2,4,2),(2,5,2),

(3,1,2),(3,2,2),(3,3,2),(3,4,2),(3,5,2)]

# без периодов

xij=[(1,1),(1,2),(1,3),(1,4),(1,5),

#второй склад

(2,1),(2,2),(2,3),(2,4),(2,5),

#третий склад

(3,1),(3,2),(3,3),(3,4),(3,5)]

#индексы количество товара поставленного от станции до складов в заданный период

yit=[(1,1),(2,1),(3,1),

# второй период

(1,2),(2,2),(3,2)]

yi1=[(1,1),(2,1),(3,1)]

yi2=[(1,2),(2,2),(3,2)]

# индексы потребности магазинов в заданный период

dtj=[(1,1),(1,2),(1,3),(1,4),(1,5),

# второй период

(2,1),(2,2),(2,3),(2,4),(2,5)]

xji=[(1,1),(2,1),(3,1),(4,1),(5,1),

# второй склад

(1,2),(2,2),(3,2),(4,2),(5,2),

# 3 склад

(1,3),(2,3),(3,3),(4,3),(5,3)]

#индексы для 1го магазина в 1 период

xi11=[(1,1,1),

#второй склад

(2,1,1),

#третий склад

(3,1,1)]

xi21=[(1,2,1),(2,2,1),(3,2,1)]

xi31=[(1,3,1),(2,3,1),(3,3,1)]

xi41=[(1,4,1),(2,4,1),(3,4,1)]

xi51=[(1,5,1),(2,5,1),(3,5,1)]

#второй период

xi12=[(1,1,2),(2,1,2),(3,1,2)]

xi22=[(1,2,2),(2,2,2),(3,2,2)]

xi32=[(1,3,2),(2,3,2),(3,3,2)]

xi42=[(1,4,2),(2,4,2),(3,4,2)]

xi52=[(1,5,2),(2,5,2),(3,5,2)]

# индексы склад 1 период

x1j1=[(1,1,1),(1,2,1),(1,3,1),(1,4,1),(1,5,1)]

x2j1=[(2,1,1),(2,2,1),(2,3,1),(2,4,1),(2,5,1)]

x3j1=[(3,1,1),(3,2,1),(3,3,1),(3,4,1),(3,5,1)]

# индексы склад 2 период

x1j2=[(1,1,2),(1,2,2),(1,3,2),(1,4,2),(1,5,2)]

x2j2=[(2,1,2),(2,2,2),(2,3,2),(2,4,2),(2,5,2)]

x3j2=[(3,1,2),(3,2,2),(3,3,2),(3,4,2),(3,5,2)]

y1t=[(1,1),(1,2)]

y2t=[(2,1),(2,2)]

y3t=[(3,1),(3,2)]

# переменные

varp1={}

for pp in xijt:

varp1[pp]=(LpVariable("x(%s,%s,%s)"%pp,lowBound=0,cat="Integer"))

varp2={}

for pp in yit:

varp2[pp]=(LpVariable("y(%s,%s)"%pp,lowBound=0,cat="Integer"))

# бинарные переменные mtj для учёта того, что на 1 и 2 складах товар расходуется полностью

varp3={}

for pp in range(1,3):

varp3[pp]=(LpVariable("m(%s)"%pp,lowBound=0,upBound=1,cat="Integer"))

# целевая функция

of=""

for (u1,v1) in xij:

of+=lpSum([varp1[(u1,v1,t)]\*v[u1-1][v1-1] for t in range(1,3)])

for u1 in range(1,4):

of+=lpSum([varp2[(u1,t)]\*c[u1-1] for t in range(1,3)])

prob+=of

for (u1,v1,z1) in xijt:

# ограничения по емкости складов

# и ограничения по потребности магазинов

prob+=varp1[(u1,v1,z1)]<=min(s[u1-1],d[z1-1][v1-1])

ss1=sum(d[0][j] for j in range(5))

ss2=sum(d[1][j] for j in range(5))

ss3=sum(s[j] for j in range(3))

for u1 in range(1,4):

prob+=varp2[(u1,1)]<=min(s[u1-1],ss1)

prob+=varp2[(u1,2)]<=min(s[u1-1],ss2)

prob+=lpSum([varp2[pp] for pp in yi1])==ss1

prob+=lpSum([varp2[pp] for pp in yi2])==ss2

prob+=lpSum([varp1[pp] for pp in xi11])==d[0][0]

prob+=lpSum([varp1[pp] for pp in xi12])==d[1][0]

prob+=lpSum([varp1[pp] for pp in xi21])==d[0][1]

prob+=lpSum([varp1[pp] for pp in xi22])==d[1][1]

prob+=lpSum([varp1[pp] for pp in xi31])==d[0][2]

prob+=lpSum([varp1[pp] for pp in xi32])==d[1][2]

prob+=lpSum([varp1[pp] for pp in xi41])==d[0][3]

prob+=lpSum([varp1[pp] for pp in xi42])==d[1][3]

prob+=lpSum([varp1[pp] for pp in xi51])==d[0][4]

prob+=lpSum([varp1[pp] for pp in xi52])==d[1][4]

prob+=lpSum([varp1[pp] for pp in x1j1])<=s[0]

prob+=lpSum([varp1[pp] for pp in x2j1])<=s[1]

prob+=lpSum([varp1[pp] for pp in x3j1])<=s[2]

prob+=lpSum([varp1[pp] for pp in x1j2])<=s[0]

prob+=lpSum([varp1[pp] for pp in x2j2])<=s[1]

prob+=lpSum([varp1[pp] for pp in x3j2])<=s[2]

prob+=lpSum([varp1[pp] for pp in x1j1])+lpSum([varp1[pp] for pp in x2j1])<=lpSum([varp1[pp] for pp in x3j1])+min(ss1,s[2])\*varp3[(1)]

prob+=lpSum([varp1[pp] for pp in x1j1])+lpSum([varp1[pp] for pp in x2j1])>=lpSum([varp1[pp] for pp in x3j1])-min(ss1,s[2])\*(1-varp3[(1)])

#второй период

prob+=lpSum([varp1[pp] for pp in x1j2])+lpSum([varp1[pp] for pp in x2j2])<=lpSum([varp1[pp] for pp in x3j2])+min(ss2,s[2])\*varp3[(2)]

prob+=lpSum([varp1[pp] for pp in x1j2])+lpSum([varp1[pp] for pp in x2j2])>=lpSum([varp1[pp] for pp in x3j2])-min(ss2,s[2])\*(1-varp3[(2)])

prob+=lpSum([varp2[pp] for pp in yi1])==lpSum([varp1[pp] for pp in x1j1])+lpSum([varp1[pp] for pp in x2j1])+lpSum([varp1[pp] for pp in x3j1])

prob+=lpSum([varp2[pp] for pp in yi2])==lpSum([varp1[pp] for pp in x1j2])+lpSum([varp1[pp] for pp in x2j2])+lpSum([varp1[pp] for pp in x3j2])

prob+=lpSum([varp2[pp] for pp in y1t])==lpSum([varp1[pp] for pp in x1j1])+lpSum([varp1[pp] for pp in x1j2])

prob+=lpSum([varp2[pp] for pp in y2t])==lpSum([varp1[pp] for pp in x2j1])+lpSum([varp1[pp] for pp in x2j2])

prob+=lpSum([varp2[pp] for pp in y3t])==lpSum([varp1[pp] for pp in x3j1])+lpSum([varp1[pp] for pp in x3j2])

prob+=varp2[(1,1)]==lpSum([varp1[pp] for pp in x1j1])

prob+=varp2[(1,2)]==lpSum([varp1[pp] for pp in x1j2])

prob+=varp2[(2,1)]==lpSum([varp1[pp] for pp in x2j1])

prob+=varp2[(2,2)]==lpSum([varp1[pp] for pp in x2j2])

prob+=varp2[(3,1)]==lpSum([varp1[pp] for pp in x3j1])

prob+=varp2[(3,2)]==lpSum([varp1[pp] for pp in x3j2])

#print(prob)

# решаем задачи линейного программирования

prob.solve()

# выводим оптимальное решение

print('Оптимальное решение:')

print(f'Общие транспортные расходы = {pulp.value(prob.objective)}')

for (i,t) in yit:

print(f'Доставлено {varp2[(i,t)].value()} единиц от станции до склада {i} в период {t}')

for (i,j,t) in xijt:

print(f'Доставлено {varp1[(i,j,t)].value()} единиц от склада {i} до объекта {j} в период {t}')

for i in range(3):

for j in range(5):

for t in range(2):

xxx1[i][j][t]=varp1[(i+1,j+1,t+1)].value()

for i in range(3):

for t in range(2):

yyy1[i][t]=varp2[(i+1,t+1)].value()

vvvv1=[7,9,14,5,1]

vvvv2=[1,9,14,5,1]

vvvv3=[17,9,14,5,1]

print(obf(xxx1,yyy1,v,c,vvvv1))

print(obf(xxx1,yyy1,v,c,vvvv2))

print(obf(xxx1,yyy1,v,c,vvvv3))

print("Изменение цены доставки от 1 склада до первого объекта не меняет общие транспортные расходы")

vvvv1=[7,9,14,5,1]

vvvv2=[7,1,14,5,1]

vvvv3=[7,19,14,5,1]

print(obf(xxx1,yyy1,v,c,vvvv1))

print(obf(xxx1,yyy1,v,c,vvvv2))

print(obf(xxx1,yyy1,v,c,vvvv3))

print("Изменение цены доставки от 1 склада до 2 объекта не меняет общие транспортные расходы")

vvvv1=[7,9,14,5,1]

vvvv2=[7,9,1,5,1]

vvvv3=[7,9,7,5,1]

print(obf(xxx1,yyy1,v,c,vvvv1))

print(obf(xxx1,yyy1,v,c,vvvv2))

print(obf(xxx1,yyy1,v,c,vvvv3))

print("Изменение цены доставки от 1 склада до 3 объекта не меняет общие транспортные расходы")

vvvv1=[7,9,14,5,1]

vvvv2=[7,9,14,1,1]

vvvv3=[7,9,14,15,1]

print(obf(xxx1,yyy1,v,c,vvvv1))

print(obf(xxx1,yyy1,v,c,vvvv2))

print(obf(xxx1,yyy1,v,c,vvvv3))

print("Изменение цены доставки от 1 склада до 4 объекта не меняет общие транспортные расходы")

vvvv1=[7,9,14,5,1]

vvvv2=[7,9,14,5,7]

vvvv3=[7,9,14,5,18]

print(obf(xxx1,yyy1,v,c,vvvv1))

print(obf(xxx1,yyy1,v,c,vvvv2))

print(obf(xxx1,yyy1,v,c,vvvv3))

print("Увеличение цены доставки от 1 склада до 5 объекта увеличивает общие транспортные расходы")

**Результат работы кода:**

Оптимальное решение:

Общие транспортные расходы = 1899.0

Доставлено 1.0 единиц от станции до склада 1 в период 1

Доставлено 0.0 единиц от станции до склада 2 в период 1

Доставлено 113.0 единиц от станции до склада 3 в период 1

Доставлено 10.0 единиц от станции до склада 1 в период 2

Доставлено 0.0 единиц от станции до склада 2 в период 2

Доставлено 145.0 единиц от станции до склада 3 в период 2

Доставлено 0.0 единиц от склада 1 до объекта 1 в период 1

Доставлено 0.0 единиц от склада 1 до объекта 2 в период 1

Доставлено 0.0 единиц от склада 1 до объекта 3 в период 1

Доставлено 0.0 единиц от склада 1 до объекта 4 в период 1

Доставлено 1.0 единиц от склада 1 до объекта 5 в период 1

Доставлено 0.0 единиц от склада 2 до объекта 1 в период 1

Доставлено 0.0 единиц от склада 2 до объекта 2 в период 1

Доставлено 0.0 единиц от склада 2 до объекта 3 в период 1

Доставлено 0.0 единиц от склада 2 до объекта 4 в период 1

Доставлено 0.0 единиц от склада 2 до объекта 5 в период 1

Доставлено 26.0 единиц от склада 3 до объекта 1 в период 1

Доставлено 12.0 единиц от склада 3 до объекта 2 в период 1

Доставлено 9.0 единиц от склада 3 до объекта 3 в период 1

Доставлено 66.0 единиц от склада 3 до объекта 4 в период 1

Доставлено 0.0 единиц от склада 3 до объекта 5 в период 1

Доставлено 0.0 единиц от склада 1 до объекта 1 в период 2

Доставлено 0.0 единиц от склада 1 до объекта 2 в период 2

Доставлено 0.0 единиц от склада 1 до объекта 3 в период 2

Доставлено 0.0 единиц от склада 1 до объекта 4 в период 2

Доставлено 10.0 единиц от склада 1 до объекта 5 в период 2

Доставлено 0.0 единиц от склада 2 до объекта 1 в период 2

Доставлено 0.0 единиц от склада 2 до объекта 2 в период 2

Доставлено 0.0 единиц от склада 2 до объекта 3 в период 2

Доставлено 0.0 единиц от склада 2 до объекта 4 в период 2

Доставлено 0.0 единиц от склада 2 до объекта 5 в период 2

Доставлено 65.0 единиц от склада 3 до объекта 1 в период 2

Доставлено 16.0 единиц от склада 3 до объекта 2 в период 2

Доставлено 55.0 единиц от склада 3 до объекта 3 в период 2

Доставлено 9.0 единиц от склада 3 до объекта 4 в период 2

Доставлено 0.0 единиц от склада 3 до объекта 5 в период 2

1899.0

1899.0

1899.0

Изменение цены доставки от 1 склада до первого объекта не меняет общие транспортные расходы

1899.0

1899.0

1899.0

Изменение цены доставки от 1 склада до 2 объекта не меняет общие транспортные расходы

1899.0

1899.0

1899.0

Изменение цены доставки от 1 склада до 3 объекта не меняет общие транспортные расходы

1899.0

1899.0

1899.0

Изменение цены доставки от 1 склада до 4 объекта не меняет общие транспортные расходы

1899.0

1965.0

2086.0

Увеличение цены доставки от 1 склада до 5 объекта увеличивает общие транспортные расходы  
  
**Вывод:**   
В данной работе были рассмотрены возможности специализированного программного обеспечения для решения задач оптимизации. Было изучено решение задач линейного программирования, решение задач динамического программирования и решение задач на оптимальное распределение с помощью языка программирования Python.