Operační výzkum I

Task 3 - Vogelova metoda a pulp knihovna

Martin Pustka

27.3.2021

1 Task 3

1.1 Zadání

Month	Production		Sales	Product 1/Product 2 Unit Cost $1,000’s		Storage
	RT	OT		RT	OT	
1	10	3	5/3	15/16	18/20	1/2
2	8	2	3/5	17/15	20/18	2/1
3	10	3	4/4	19/17	22/22	

1.2 Zpracované zadání

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand
M1P1	15	18	M	M	M	M	5
M1P2	16	20	M	M	M	M	3
M2P1	16	19	17	20	M	M	3
M2P2	17	22	15	18	M	M	5
M3P1	18	21	19	22	19	22	4
M3P2	18	23	16	19	17	22	4
Dummy	0	0	0	0	0	0	12
Supply	10	3	8	2	10	3	

1.3 Vogelova aproximační metoda

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand	Diff
M1P1	15	18	M	M	M	M	5	3
M1P2	16	20	M	M	M	M	3	4
M2P1	16	19	17	20	M	M	3	1
M2P2	17	22	15	18	M	M	5	2
M3P1	18	21	19	22	19	22	4	1
M3P2	18	23	16	19	17	22	4	1
Dummy	0	0	0	0	0	0	12	0
Supply	10	3	8	2	10	3	x76=3	
Diff	15	18	15	18	17	22		

ze sloupečku nebo řádku Diff vybereme největší hodnotu
- z řádku vybráno 22
v příslušném sloupečku nebo řádku vybereme nejmenší hodnotu
- ze sloupečku vybráno 0 na pozici x76
vybereme menší ze dvou hodnot ve sloupci Demand nebo Supply
- ze Supply vybráno 3
zrušíme řádek nebo sloupec a snížíme hodnotu v Demand nebo Supply
- při vybrání hodnoty Supply rušíme sloupec a v řádku dosadíme 12-3 = 9

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand	Diff
M1P1	15	18	M	M	M	x	5	3
M1P2	16	20	M	M	M	x	3	4
M2P1	16	19	17	20	M	x	3	1
M2P2	17	22	15	18	M	x	5	2
M3P1	18	21	19	22	19	x	4	1
M3P2	18	23	16	19	17	x	4	1
Dummy	0	0	0	0	0	x	9	0
Supply	10	3	8	2	10	x	x74=2	
Diff	15	18	15	18	17	x		

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand	Diff
M1P1	15	18	M	x	M	x	5	3
M1P2	16	20	M	x	M	x	3	4
M2P1	16	19	17	x	M	x	3	1
M2P2	17	22	15	x	M	x	5	2
M3P1	18	21	19	x	19	x	4	1
M3P2	18	23	16	x	17	x	4	1
Dummy	0	0	0	x	0	x	7	0
Supply	10	3	8	x	10	x	x72=3	
Diff	15	18	15	x	17	x		

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand	Diff
M1P1	15	x	M	x	M	x	5	15
M1P2	16	x	M	x	M	x	3	16
M2P1	16	x	17	x	M	x	3	1
M2P2	17	x	15	x	M	x	5	2
M3P1	18	x	19	x	19	x	4	1
M3P2	18	x	16	x	17	x	4	1
Dummy	0	x	0	x	0	x	4	0
Supply	10	x	8	x	10	x	x75=4	
Diff	15	x	15	x	17	x		

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand	Diff
M1P1	15	x	M	x	M	x	5	15
M1P2	16	x	M	x	M	x	3	16
M2P1	16	x	17	x	M	x	3	1
M2P2	17	x	15	x	M	x	5	2
M3P1	18	x	19	x	19	x	4	1
M3P2	18	x	16	x	17	x	4	1
Dummy	x	x	x	x	x	x	x	x
Supply	10	x	8	x	6	x	x21=3	
Diff	1	x	1	x	2	x		

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand	Diff
M1P1	15	x	M	x	M	x	5	15
M1P2	x	x	x	x	x	x	x	x
M2P1	16	x	17	x	M	x	3	1
M2P2	17	x	15	x	M	x	5	2
M3P1	18	x	19	x	19	x	4	1
M3P2	18	x	16	x	17	x	4	1
Dummy	x	x	x	x	x	x	x	x
Supply	7	x	8	x	6	x	x11=5	
Diff	1	x	1	x	2	x		

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand	Diff
M1P1	x	x	x	x	x	x	x	x
M1P2	x	x	x	x	x	x	x	x
M2P1	16	x	17	x	M	x	3	1
M2P2	17	x	15	x	M	x	5	2
M3P1	18	x	19	x	19	x	4	1
M3P2	18	x	16	x	17	x	4	1
Dummy	x	x	x	x	x	x	x	x
Supply	2	x	8	x	6	x	x43=5	
Diff	1	x	1	x	2	x		

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand	Diff
M1P1	x	x	x	x	x	x	x	x
M1P2	x	x	x	x	x	x	x	x
M2P1	16	x	17	x	M	x	3	1
M2P2	x	x	x	x	x	x	x	x
M3P1	18	x	19	x	19	x	4	1
M3P2	18	x	16	x	17	x	4	1
Dummy	x	x	x	x	x	x	x	x
Supply	2	x	3	x	6	x	x31=2	
Diff	2	x	1	x	2	x		

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand	Diff
M1P1	x	x	x	x	x	x	x	x
M1P2	x	x	x	x	x	x	x	x
M2P1	x	x	17	x	M	x	1	17
M2P2	x	x	x	x	x	x	x	x
M3P1	x	x	19	x	19	x	4	0
M3P2	x	x	16	x	17	x	4	1
Dummy	x	x	x	x	x	x	x	x
Supply	x	x	3	x	6	x	x33=1	
Diff	x	x	1	x	2	x		

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand	Diff
M1P1	x	x	x	x	x	x	x	x
M1P2	x	x	x	x	x	x	x	x
M2P1	x	x	x	x	x	x	x	x
M2P2	x	x	x	x	x	x	x	x
M3P1	x	x	19	x	19	x	4	0
M3P2	x	x	16	x	17	x	4	1
Dummy	x	x	x	x	x	x	x	x
Supply	x	x	2	x	6	x	x63=2	
Diff	x	x	3	x	2	x		

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand	Diff
M1P1	x	x	x	x	x	x	x	x
M1P2	x	x	x	x	x	x	x	x
M2P1	x	x	x	x	x	x	x	x
M2P2	x	x	x	x	x	x	x	x
M3P1	x	x	x	x	19	x	4	19
M3P2	x	x	x	x	17	x	2	17
Dummy	x	x	x	x	x	x	x	x
Supply	x	x	x	x	6	x	x55=4	
Diff	x	x	x	x	2	x		

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand	Diff
M1P1	x	x	x	x	x	x	x	x
M1P2	x	x	x	x	x	x	x	x
M2P1	x	x	x	x	x	x	x	x
M2P2	x	x	x	x	x	x	x	x
M3P1	x	x	x	x	x	x	x	x
M3P2	x	x	x	x	17	x	2	17
Dummy	x	x	x	x	x	x	x	x
Supply	x	x	x	x	2	x	x65=2	
Diff	x	x	x	x	17	x		

1.4 Kontrola

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	u values
M1P1	15=r1+s1	x	x	x	x	x	17
M1P2	16=r2+s1	x	x	x	x	x	18
M2P1	16=r3+s1	x	17=r3+s3	x	x	x	18
M2P2	x	x	15=r4+s3	x	x	x	16
M3P1	x	x	x	x	19=r5+s5	x	19
M3P2	x	x	16=r6+s3	x	17=r6+s5	x	17
Dummy	x	0=r7+s2	x	0=r7+s4	0=r7+s5	0=r7+s6	0
v values	-2	0	-1	0	0	0	

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT
M1P1	15	x	x	x	x	x
M1P2	16	x	x	x	x	x
M2P1	16	x	17	x	x	x
M2P2	x	x	15	x	x	x
M3P1	x	x	x	x	19	x
M3P2	x	x	16	x	17	x
Dummy	x	0	x	0	0	0

hodnoty, které jsou podstatné

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT
M1P1	0	x	x	x	x	x
M1P2	0	x	x	x	x	x
M2P1	0	x	0	x	x	x
M2P2	x	x	0	x	x	x
M3P1	x	x	x	x	0	x
M3P2	x	x	0	x	0	x
Dummy	x	0	x	0	0	0

všechny hodnoty jsou nezáporné, nalezeno optimální řešení

1.5 Výsledky

1.5.1 Vogelova metoda

```
x76 = 3
x74 = 2
x72 = 3
x75 = 4
x21 = 3
x11 = 5
x43 = 5
x31 = 2
x33 = 1
x63 = 2
x55 = 4
x65 = 2
Total costs = 389.0
```

1.5.2 Python pulp knihovna

```
x11 = 5.0
x21 = 3.0
x31 = 2.0
x33 = 1.0
x43 = 5.0
x55 = 4.0
x63 = 2.0
x65 = 2.0
x72 = 3.0
x74 = 2.0
x75 = 4.0
x76 = 3.0
Total costs = 389.0
```

1.5.3 Převedení do tabulky

	M1RT	M1OT	M2RT	M2OT	M3RT	M3OT	Demand
M1P1	5	0	0	0	0	0	5
M1P2	3	0	0	0	0	0	3
M2P1	2	0	1	0	0	0	3
M2P2	0	0	5	0	0	0	5
M3P1	0	0	0	0	4	0	4
M3P2	0	0	2	0	2	0	4
Dummy	0	3	0	2	4	3	12
Supply	10	3	8	2	10	3	

1.6 Python zdrojový kód

```
from pulp import *
import numpy as np

#Task 3

Factories = [”1”, ”2”, ”3”, ”4”, ”5”, ”6”]

supply = {”1”: 10, ”2”: 3, ”3”: 8, ”4”: 2, ”5”:10, ”6”:3}

Products = [”1”, ”2”, ”3”, ”4”, ”5”, ”6”, ”7”]

demand = {”1”: 5, ”2”:3, ”3”:3, ”4”:5, ”5”:4, ”6”:4, ”7”:12}

costs = {”1”:{”1”:15, ”2”: 16, ”3”: 16, ”4”: 17, ”5”: 18, ”6”: 18, ”7”: 0 },
         ”2”:{”1”:18, ”2”: 20, ”3”: 19, ”4”: 22, ”5”: 21, ”6”: 23, ”7”: 0 },
         ”3”:{”1”:1e6, ”2”: 1e6, ”3”: 17, ”4”: 15, ”5”: 19, ”6”: 16, ”7”: 0 },
         ”4”:{”1”:1e6, ”2”: 1e6, ”3”: 20, ”4”: 18, ”5”: 22, ”6”: 19, ”7”: 0 },
         ”5”:{”1”:1e6, ”2”: 1e6, ”3”: 1e6, ”4”: 1e6, ”5”: 19, ”6”: 17, ”7”: 0 },
         ”6”:{”1”:1e6, ”2”: 1e6, ”3”: 1e6, ”4”: 1e6, ”5”: 22, ”6”: 22, ”7”: 0 }
         }

prob = LpProblem(”Factories and Products”, LpMinimize)

Routes = [(f,p) for f in Factories for p in Products]

route_vars = LpVariable.dicts(”Route”, (Factories, Products), 0, None, LpInteger)

prob += lpSum([route_vars[f][p]*costs[f][p] for (f,p) in Routes]), ”Sum of Transporting Costs”

# The supply maximum constraints are added to prob for each supply node (warehouse)
for f in Factories:
    prob += lpSum([route_vars[f][p] for p in Products]) <= supply[f], ”Sum of Products out of Plants %s”%f

# The demand minimum constraints are added to prob for each demand node (bar)
for p in Products:
    prob += lpSum([route_vars[f][p] for f in Factories]) >= demand[p], ”Sum of Products into Warehouses %s”%p

rts = []

prob.solve()
for v in prob.variables():
    if(v.varValue > 0):
        x = v.name.split(”_”)
        rts.append(f”x{x[2]}{x[1]} = {v.varValue}”)

rts.sort()
for route in rts:
    print(route)

print('Total costs = ', value(prob.objective))
```