

Sprawozdanie: Wektorowanie statków powietrznych przy zastosowaniu metody programowania matematycznego, cz.2

Wojciech Chwaciński 151924

Kod źródłowy:

```
import re
from docplex.mp.model import Model
import time
file_name = "cm\CM_n=40_m=7.txt"
numbers = re.findall(r'\d+', file_name)
print(numbers)
file = open(file_name, 'r', encoding='Utf-8')

matrix = []
n = int(numbers[0]) # liczba samolotów
m = int(numbers[1]) # liczba manewrów
print(n, m)

for i in range(n*m):
    read = file.readline()
    num = re.findall(r'\d+', read)
    dane = list(map(int, num))
    matrix.append(dane)
print(matrix)

model = Model()
model.populate_solution_pool = True

x = {(i, j): model.binary_var(name=f'x_{i}_{j}') for i in range(n) for j in range(m)}

#for i in range(n):
#    model.add_constraint(model.sum(x[i, j] for j in range(m) ) == 1,
#                           ctname=f'sum_row_{i}')
for i in range(n):
    model.add_constraint(x[i, 0]+x[i, 1]+x[i, 2]+x[i, 3]+x[i, 4]+x[i, 5]+x[i,6] == 1)

flaga = 0
for i in range(len(matrix)):
    if i%m == 0:
        flaga += 1
    for j in range(flaga*m, len(matrix[i])):
        if matrix[i][j] == 1 and i != j:
            print(i,j,"\n")
            nr_pierwszego_samolotu = int(i/m)
```

```

        nr_pierwszego_manewru = j
        while nr_pierwszego_manewru > (m-1):
            nr_pierwszego_manewru = nr_pierwszego_manewru - m

        nr_drugiego_samolotu = int(j/m)

        nr_drugiego_manewru = i
        while nr_drugiego_manewru > (m-1):
            nr_drugiego_manewru = nr_drugiego_manewru - m

        print(f"pierwsza zmienna: samolot nr: {nr_pierwszego_samolotu}
manewr nr: {nr_pierwszego_manewru}")
        print(f"druga zmienna: samolot nr: {nr_drugiego_samolotu} manewr
nr: {nr_drugiego_manewru}")
        model.add_constraint(x[nr_pierwszego_samolotu,
nr_pierwszego_manewru] + x[nr_drugiego_samolotu, nr_drugiego_manewru] <= 1)

objective_expr = model.sum(x[i, j] * (j+1) for i in range(0, n) for j in
range(0, m))
model.minimize(objective_expr)

start = time.time()
solution = model.solve()
end = time.time()
print(f"Czas potrzebny na obliczenia: {end-start}[s]")
#print(solution)
#print(x)

print(f"Liczba znalezionych rozwiązań dopuszczalnych:
{model.populate().size}")

if solution is not None:
    print("Przykładowe rozwiązanie\n")
    for i in range(n):
        for j in range(m):
            if x[i,j].solution_value == 1:
                print(f'Samolot {i}, manewr {j}: {x[i, j].solution_value}')
else:
    print('Brak rozwiązania.')

```

Wyniki dla CM_n=40_m=7.txt:

Rozwiązanie-

Samolot 0, manewr 1: 1.0

Samolot 1, manewr 1: 1.0

Samolot 2, manewr 0: 1.0

Samolot 3, manewr 3: 1.0

Samolot 4, manewr 2: 1.0

Samolot 5, manewr 3: 1.0

Samolot 6, manewr 1: 1.0

Samolot 7, manewr 0: 1.0

Samolot 8, manewr 2: 1.0

Samolot 9, manewr 2: 1.0

Samolot 10, manewr 3: 1.0

Samolot 11, manewr 0: 1.0

Samolot 12, manewr 2: 1.0

Samolot 13, manewr 1: 1.0

Samolot 14, manewr 0: 1.0

Samolot 15, manewr 1: 1.0

Samolot 16, manewr 0: 1.0

Samolot 17, manewr 4: 1.0

Samolot 18, manewr 5: 1.0

Samolot 19, manewr 0: 1.0

Samolot 20, manewr 0: 1.0

Samolot 21, manewr 3: 1.0

Samolot 22, manewr 2: 1.0

Samolot 23, manewr 5: 1.0

Samolot 24, manewr 0: 1.0

Samolot 25, manewr 2: 1.0

Samolot 26, manewr 2: 1.0

Samolot 27, manewr 3: 1.0

Samolot 28, manewr 1: 1.0

Samolot 29, manewr 0: 1.0

Samolot 30, manewr 1: 1.0

Samolot 31, manewr 0: 1.0

Samolot 32, manewr 0: 1.0

Samolot 33, manewr 1: 1.0

Samolot 34, manewr 0: 1.0

Samolot 35, manewr 2: 1.0

Samolot 36, manewr 3: 1.0

Samolot 37, manewr 3: 1.0

Samolot 38, manewr 2: 1.0

Samolot 39, manewr 4: 1.0

Czas obliczeń:

1.6853976249694824[s]

(czas z poprzedniego zadania 0.0169980525970459[s])

Wyniki dla CM_n=30_m=7.txt:

Rozwiązanie-

Samolot 0, manewr 1: 1.0

Samolot 1, manewr 0: 1.0

Samolot 2, manewr 2: 1.0

Samolot 3, manewr 3: 1.0

Samolot 4, manewr 2: 1.0

Samolot 5, manewr 0: 1.0

Samolot 6, manewr 0: 1.0

Samolot 7, manewr 0: 1.0

Samolot 8, manewr 2: 1.0

Samolot 9, manewr 0: 1.0

Samolot 10, manewr 1: 1.0

Samolot 11, manewr 1: 1.0

Samolot 12, manewr 2: 1.0

Samolot 13, manewr 0: 1.0
Samolot 14, manewr 1: 1.0
Samolot 15, manewr 2: 1.0
Samolot 16, manewr 0: 1.0
Samolot 17, manewr 1: 1.0
Samolot 18, manewr 1: 1.0
Samolot 19, manewr 0: 1.0
Samolot 20, manewr 1: 1.0
Samolot 21, manewr 2: 1.0
Samolot 22, manewr 1: 1.0
Samolot 23, manewr 0: 1.0
Samolot 24, manewr 1: 1.0
Samolot 25, manewr 0: 1.0
Samolot 26, manewr 4: 1.0
Samolot 27, manewr 0: 1.0
Samolot 28, manewr 0: 1.0
Samolot 29, manewr 3: 1.0

Czas obliczeń:

0.16318750381469727[s]

(czas z poprzedniego zadania 0.012999534606933594[s])

Wyniki dla CM_n=20_m=7.txt:

Rozwiązanie-

Samolot 0, manewr 0: 1.0
Samolot 1, manewr 2: 1.0
Samolot 2, manewr 1: 1.0
Samolot 3, manewr 0: 1.0
Samolot 4, manewr 0: 1.0
Samolot 5, manewr 0: 1.0
Samolot 6, manewr 0: 1.0

Samolot 7, manewr 0: 1.0
Samolot 8, manewr 0: 1.0
Samolot 9, manewr 2: 1.0
Samolot 10, manewr 0: 1.0
Samolot 11, manewr 0: 1.0
Samolot 12, manewr 0: 1.0
Samolot 13, manewr 1: 1.0
Samolot 14, manewr 0: 1.0
Samolot 15, manewr 0: 1.0
Samolot 16, manewr 0: 1.0
Samolot 17, manewr 1: 1.0
Samolot 18, manewr 2: 1.0
Samolot 19, manewr 0: 1.0

Czas obliczeń:

0.03931903839111328[s]

(czas z poprzedniego zadania 0.011998176574707031[s])

Wyniki dla CM_n=10_m=7.txt:

Rozwiązanie-

Samolot 0, manewr 0: 1.0
Samolot 1, manewr 1: 1.0
Samolot 2, manewr 0: 1.0
Samolot 3, manewr 2: 1.0
Samolot 4, manewr 1: 1.0
Samolot 5, manewr 0: 1.0
Samolot 6, manewr 0: 1.0
Samolot 7, manewr 0: 1.0
Samolot 8, manewr 0: 1.0
Samolot 9, manewr 0: 1.0

Czas obliczeń:

0.01875925064086914[s]

(czas z poprzedniego zadania 0.007997989654541016[s])

Wnioski:

Czasy pracy solvera wydłużyły się w każdym przypadku zaczynając od 10 samolotów czas wzrósł mniej więcej dwukrotnie, przy 20 samolotach szukanie wyniku optymalnego zajęło 3 razy więcej czasu, następnie przy 30 czas wydłużył się ponad 12 razy a patrząc na ostateczne wyniki obecny program potrzebował ponad 100 razy więcej czasu od swojego poprzednika na wykonanie obliczeń. Czas obliczeń wydłużył się znacznie ze względu na dużo większą złożoność problemu gdyż poszukiwanie rozwiązania optymalnego jest o wiele trudniejszym zadaniem niż odnalezienie rozwiązania dopuszczalnego a staje się tym trudniejsze im większa jest instancja wejściowa.