

# Sprawozdanie 2

Problem komiwojażera LINGO

## Spis treści

Opis ogólny .....	2
Zadanie 1 .....	2
Zadanie 2 .....	2
Zadanie 3 .....	3
Podsumowanie .....	4

## Opis ogólny

Podczas przygotowywania sprawozdania korzystałem z oprogramowania LINGO 18.0. Zadanie polegało na modyfikacji przedstawionego na zajęciach przykładowego pliku *TSP.lg4* wbudowanego w używany program. Posiadając nieparzysty numer indeksu (254279), mój problem musiał być metryczny i niesymetryczny.

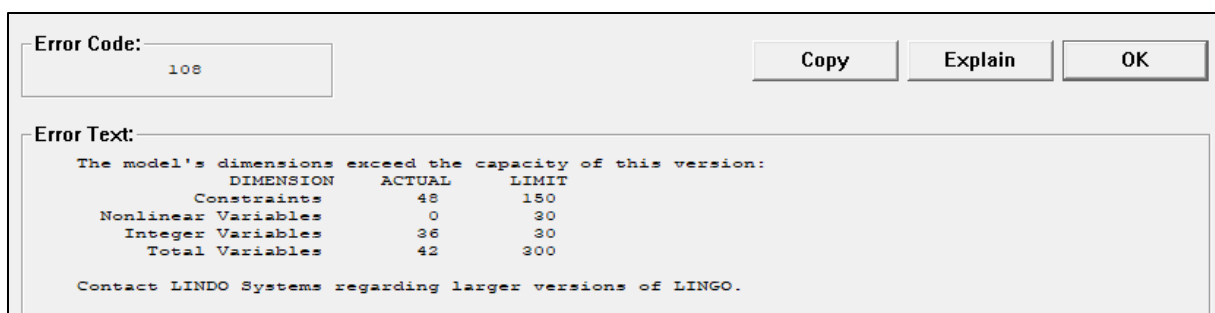
## Zadanie 1

Pierwszym krokiem było niejako spersonalizowanie programu modyfikując dane wejściowe na takie, które byłyby powiązane z numerem mojego indeksu. Czynność tę wykonałem w następujący sposób:

```
DATA:  !Niesymetryczna macierz kosztów transportu;
COST =  0  254  279  254  279
        432   0  452  413  312
        381  411   0  311  380
        342  393  401   0  319
        254  279  254  279   0;
ENDDATA
```

Pierwsza oraz ostatnia linijka wprowadzonych danych to podwójnie wpisany mój numer indeksu podzielony na fragmenty 3-cyfrowe.

Program w wersji testowej pozwolił na wprowadzenie maksymalnie 5 miast oraz przeprowadzenie dla nich obliczeń. Przy większej ilości widoczny jest poniższy komunikat:



## Zadanie 2

Problem komiwojażera według polecenia powinien być metryczny. Znaczy to, że dane wejściowe muszą spełniać nierówność trójkąta. Możemy sprawdzić to biorąc dwie najmniejsze wartości i sprawdzić czy po dodaniu są one większe od największej wartości w macierzy kosztów. Nie bierzemy tutaj pod uwagę wartości na przekątnej, ponieważ informuje ona o koszcie przewozu towaru z miasta A do miasta A, a takiego przypadku nie bierzemy pod uwagę.

$$254 + 279 > 452$$

```
DATA:      !Niesymetryczna macierz kosztów transportu;
COST =    0  254  279  254  279
          432   0  452  413  312
          381  411   0  311  380
          342  393  401   0  319
          254  279  254  279   0;
ENDDATA
```

Zielonym podkreśleniem zaznaczone są najmniejsze wartości, niebieskim wartość największa.

### Zadanie 3

Posiadając nieparzysty numer indeksu problem mój powinien być także niesymetryczny. W macierzy kosztów przekątna wyznacza oś symetrii. Wprowadzając dane musiałem zadbać, by na przykład koszt przewozu towaru z miasta 2 do miasta 5 była różna od kosztu transportu z miasta 5 do miasta 2.

$$279 \neq 312$$

Poniżej wspomniana różnica zaznaczona na żółto:

```
DATA:      !Niesymetryczna macierz kosztów transportu;
COST =    0  254  279  254  279
          432   0  452  413  312
          381  411   0  311  380
          342  393  401   0  319
          254  279  254  279   0;
ENDDATA
```

## Podsumowanie

Implementacja całego programu wygląda w następujący sposób:

```

MODEL:
  ! Problem komiwojażera;
SETS:
  CITY / 1.. 5/: U;
  LINK( CITY, CITY):
    COST, ! Macierz kosztów;
    X;
ENDSETS
DATA: !Niesymetryczna macierz kosztów transportu;
  COST =
    0 254 279 254 279
    432 0 452 413 312
    381 411 0 311 380
    342 393 401 0 319
    254 279 254 279 0;
ENDDATA

N = @SIZE( CITY);
MIN = @SUM( LINK: COST * X);
@FOR( CITY( K):
  @SUM( CITY( I) | I #NE# K: X( I, K)) = 1;
  @SUM( CITY( J) | J #NE# K: X( K, J)) = 1;
  @FOR( CITY( J) | J #GT# 1 #AND# J #NE# K:
    U( J) >= U( K) + X( K, J) -
      ( N - 2) * ( 1 - X( K, J)) +
      ( N - 3) * X( J, K)
  );
);
! Zmienna binarna;
@FOR( LINK: @BIN( X));
@FOR( CITY( K) | K #GT# 1:
  U( K) <= N - 1 - ( N - 2) * X( 1, K);
  U( K) >= 1 + ( N - 2) * X( K, 1)
);
END

```

Otrzymany finalny wynik przedstawiam poniżej:

```
Global optimal solution found.
Objective value:                1473.000
Objective bound:                1473.000
Infeasibilities:                0.000000
Extended solver steps:          0
Total solver iterations:        12
Elapsed runtime seconds:        0.07

Model Class:                    MILP

Total variables:                30
Nonlinear variables:            0
Integer variables:              25

Total constraints:              35
Nonlinear constraints:          0

Total nonzeros:                140
Nonlinear nonzeros:            0
```

Variable	Value	Reduced Cost
N	5.000000	0.000000
U( 1)	0.000000	0.000000
U( 2)	1.000000	0.000000
U( 3)	3.000000	0.000000
U( 4)	4.000000	0.000000
U( 5)	2.000000	0.000000
COST( 1, 1)	0.000000	0.000000
COST( 1, 2)	254.0000	0.000000
COST( 1, 3)	279.0000	0.000000
COST( 1, 4)	254.0000	0.000000
COST( 1, 5)	279.0000	0.000000
COST( 2, 1)	432.0000	0.000000
COST( 2, 2)	0.000000	0.000000
COST( 2, 3)	452.0000	0.000000
COST( 2, 4)	413.0000	0.000000
COST( 2, 5)	312.0000	0.000000
COST( 3, 1)	381.0000	0.000000
COST( 3, 2)	411.0000	0.000000
COST( 3, 3)	0.000000	0.000000
COST( 3, 4)	311.0000	0.000000
COST( 3, 5)	380.0000	0.000000
COST( 4, 1)	342.0000	0.000000
COST( 4, 2)	393.0000	0.000000
COST( 4, 3)	401.0000	0.000000
COST( 4, 4)	0.000000	0.000000
COST( 4, 5)	319.0000	0.000000
COST( 5, 1)	254.0000	0.000000
COST( 5, 2)	279.0000	0.000000
COST( 5, 3)	254.0000	0.000000
COST( 5, 4)	279.0000	0.000000
COST( 5, 5)	0.000000	0.000000

X( 1, 1)	0.000000	0.000000
X( 1, 2)	1.000000	254.0000
X( 1, 3)	0.000000	279.0000
X( 1, 4)	0.000000	254.0000
X( 1, 5)	0.000000	279.0000
X( 2, 1)	0.000000	432.0000
X( 2, 2)	0.000000	0.000000
X( 2, 3)	0.000000	452.0000
X( 2, 4)	0.000000	413.0000
X( 2, 5)	1.000000	312.0000
X( 3, 1)	0.000000	381.0000
X( 3, 2)	0.000000	411.0000
X( 3, 3)	0.000000	0.000000
X( 3, 4)	1.000000	311.0000
X( 3, 5)	0.000000	380.0000
X( 4, 1)	1.000000	342.0000
X( 4, 2)	0.000000	393.0000
X( 4, 3)	0.000000	401.0000
X( 4, 4)	0.000000	0.000000
X( 4, 5)	0.000000	319.0000
X( 5, 1)	0.000000	254.0000
X( 5, 2)	0.000000	279.0000
X( 5, 3)	1.000000	254.0000
X( 5, 4)	0.000000	279.0000
X( 5, 5)	0.000000	0.000000
Row	Slack or Surplus	Dual Price
1	0.000000	0.000000
2	1473.000	-1.000000
3	0.000000	0.000000
4	0.000000	0.000000
5	0.000000	0.000000
6	6.000000	0.000000
7	5.000000	0.000000
8	5.000000	0.000000
9	0.000000	0.000000
10	0.000000	0.000000
11	5.000000	0.000000
12	6.000000	0.000000
13	0.000000	0.000000
14	0.000000	0.000000
15	0.000000	0.000000
16	1.000000	0.000000
17	0.000000	0.000000
18	0.000000	0.000000
19	0.000000	0.000000
20	0.000000	0.000000
21	0.000000	0.000000
22	0.000000	0.000000
23	1.000000	0.000000
24	0.000000	0.000000
25	0.000000	0.000000
26	0.000000	0.000000
27	0.000000	0.000000
28	5.000000	0.000000
29	0.000000	0.000000
30	0.000000	0.000000
31	1.000000	0.000000
32	2.000000	0.000000
33	0.000000	0.000000
34	0.000000	0.000000
35	2.000000	0.000000
36	1.000000	0.000000