Bernhard C. Schrenk <clemy@clemy.org>

Hi,

I found your challenge as a colleague mentioned it to me and I thought I will give it a try. I used integer linear programming to model it and solved it with Gurobi finding 2 solutions:

```
Solution 1:
Kasparov    14
Karpov      13
Fischer     15
Solution 2:
Kasparov    13
Karpov      14
Fischer     15
```

My model:

```
Minimize
 ks_ka_w + ks_ka_d + ks_ka_l + ks_fi_w + ks_fi_d + ks_fi_l + ka_fi_w +
ka_fi_d + ka_fi_l

Subject To
 c0: ks_ka_w + ks_ka_d + ks_ka_l = 7
 c1: ks_fi_w + ks_fi_d + ks_fi_l = 7
 c2: ka_fi_w + ka_fi_d + ka_fi_l = 7
 c3: ks_ka_w + ks_fi_w - ks_ka_l - ka_fi_w >= 1
 c4: ks_ka_w + ks_fi_w - ks_fi_l - ka_fi_l >= 1
 c5: ks_ka_w + ka_fi_l - ks_ka_l - ks_fi_l <= -1
 c6: ks_ka_w + ka_fi_l - ks_fi_w - ka_fi_w <= -1
 c7: ks_points - 2 ks_ka_w - ks_ka_d - 2 ks_fi_w - ks_fi_d = 0
 c8: ka_points - 2 ks_ka_l - ks_ka_d - 2 ka_fi_w - ka_fi_d = 0
 c9: fi_points - 2 ks_fi_l - ks_fi_d - 2 ka_fi_l - ka_fi_d = 0
 c10: fi_points - ks_points >= 1
 c11: fi_points - ka_points >= 1

General
 ks_ka_w ks_ka_d ks_ka_l ks_fi_w ks_fi_d ks_fi_l ka_fi_w ka_fi_d
ka_fi_l ks_points ka_points fi_points

End
```

My abbreviations for variables:

```
ks..Kasparov
ka..Karpov
fi..Fischer
w..Win
d..Draw
l..Loose
```

`ks_ka_w` is therefore the number of wins of Kasparov against Karpov (which is the same as looses of Karpov against Kasparov), ...
`ks_points` are the tournament points of Kasparov,..

The constraints c0-c2 encode the number of games played.
The constraints c3,c4 encode that Kasparov has the most wins.
The constraints c5,c6 encode that Karpov has the least looses.
The constraints c7-c9 are just for calculating the points (to have them in the output).
The constraints c10,c11 encode that Fischer wins the tournament.

To run the model from above (stored in "chess.lp") in Gurobi and get all possible solutions, following python script is used:

```
import gurobipy as gp
m = gp.read("chess.lp")
m.Params.PoolSearchMode = 2
m.optimize()
for i in range(m.SolCount):
    m.Params.SolutionNumber = i
    m.printAttr("xn")
```

You can find the full output below. Have fun with my solution.

Regards,
Bernhard Clemens Schrenk

```
Read LP format model from file chess.lp
Reading time = 0.01 seconds
: 12 rows, 12 columns, 44 nonzeros
Changed value of parameter PoolSearchMode to 2
   Prev: 0  Min: 0  Max: 2  Default: 0
Gurobi Optimizer version 9.0.3 build v9.0.3rc0 (win64)
Optimize a model with 12 rows, 12 columns and 44 nonzeros
Model fingerprint: 0x92f8f182
Variable types: 0 continuous, 12 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 2e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+00, 7e+00]
Presolve removed 2 rows and 2 columns
Presolve time: 0.00s
Presolved: 10 rows, 10 columns, 40 nonzeros
Variable types: 0 continuous, 10 integer (0 binary)

Root relaxation: objective 2.100000e+01, 10 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective
Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap |
It/Node Time
```

```
*    0     0               0      21.0000000   21.00000  0.00%     -
 0s

Optimal solution found at node 0 - now completing solution pool...

    Nodes    |    Current Node    |      Pool Obj.
Bounds     |     Work
             |                    |   Worst                    |
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap |
It/Node Time

     0     0          -    0               -   21.00000      -     -
 0s
     0     0          -    0               -   21.00000      -     -
 0s
     0     2          -    0               -   21.00000      -     -
 0s

Explored 11 nodes (16 simplex iterations) in 0.03 seconds
Thread count was 4 (of 4 available processors)

Solution count 2: 21 21
No other solutions better than 1e+100

Optimal solution found (tolerance 1.00e-04)
Best objective 2.100000000000e+01, best bound 2.100000000000e+01, gap
0.0000%

    Variable           xn
-------------------------
     ks_ka_w            2
     ks_ka_d            4
     ks_ka_l            1
     ks_fi_w            3
     ks_fi_l            4
     ka_fi_d            7
   ks_points           14
   ka_points           13
   fi_points           15

    Variable           xn
-------------------------
     ks_ka_w            2
     ks_ka_d            3
     ks_ka_l            2
     ks_fi_w            3
     ks_fi_l            4
     ka_fi_d            7
   ks_points           13
   ka_points           14
   fi_points           15
```