Mathematics of Infrastructure Planning

The thirty-six officers problem

Stefan Heinz

Zuse Institute Berlin (ZIB)

DFG Research Center MATHEON
Mathematics for key technologies

19/April/2012

ZIB Optimization Suite = SCIP + SoPlex + ZIMPL

Toolbox for generating and solving constraint integer programs

ZIMPL

- a mixed integer programming modeling language
- easily generate LPs, MIPs, and ...

SCIP

- a MIP and CP solver, branch-cut-and-price framework
- ZIMPL models can directly be loaded into SCIP and solved

SoPlex

- a linear programming solver
- SCIP uses SoPlex as underlying LP solver

The thirty-six officers problem

Problem description

This problem asks for an arrangement of 36 officers of 6 ranks and from 6 regiments in a square formation of size 6 by 6. Each vertical and each horizontal line of this formation is to contain one and only one officer of each rank and one and only one officer from each regiment.

Reference

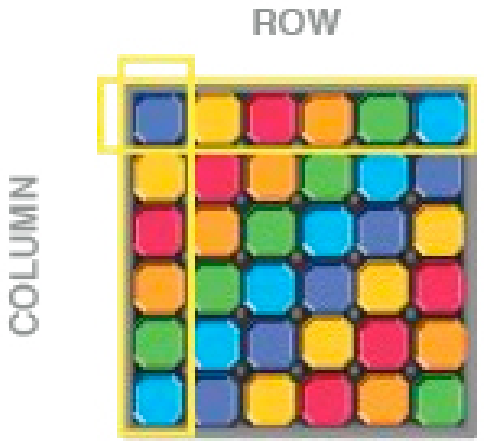
Leonhard Euler, "Recherches sur une nouvelle espce de quarrs magiques" Verhandelingen uitgegeven door het zeeuwsch Genootschap der Wetenschappen te Vlissingen 9, Middelburg 1782, pp. 85–239

`http://www.math.dartmouth.edu/~euler/pages/E530.html`

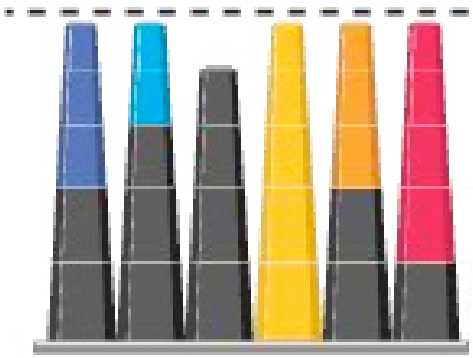
Translation

`http://www.math.dartmouth.edu/~euler/docs/translations/E530.pdf`

36 Cube



Place one of each colour in every row and column.



Towers must fit to form a level cube.

36 Towers
Genius!

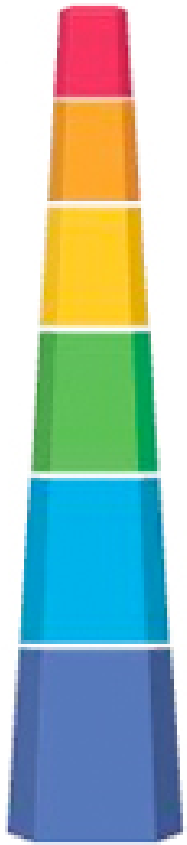
35 Towers
Liar!

31-34 Towers
Rockstar!

26-30 Towers
Contender

20-25 Towers
Varsity

19 or Fewer Towers
Novice



6

1	2	5	4	6	3
5	3	6	1	4	2
4	6	3	5	2	1
2	1	4	3	5	6
3	5	2	6	1	4
6	4	1	2	3	5

ZIMPL Model (Parsing)

```
# board size
param size := read "36cube.dat" as "1n" use 1;
do print "size = ", size;

# create sets

# we have 6 different heights
set Heights := {1..size};

# we have 6 different colors
set Colors := {1..size};

# there are 6 rows and 6 columns
set Rows := {1..size};
set Columns := {1..size};

# parse heights
param heights[<r,c> in Rows * Columns] :=
      read "36cube.dat" as "<1n,2n> 3n" skip 1;
#do forall <r,c> in Rows * Columns do print heights[r,c];
```

ZIMPL Model (Decision variables)

```
# binary variables to define the setup
# z defines in which color goes on which position
var z[Rows * Columns * Colors] binary;
```

ZIMPL Model (Constraints)

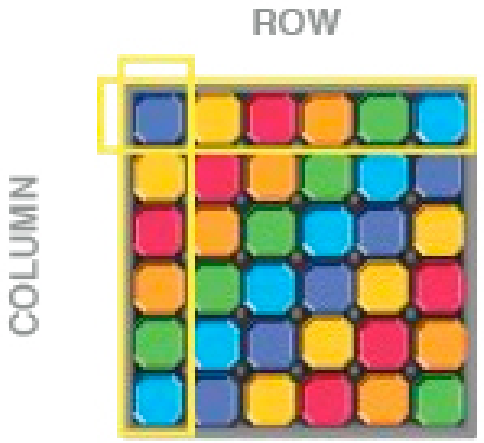
```
# each position gets one color
subto color :
   forall <r> in Rows:
   forall <c> in Columns :
      sum <i> in Colors : z[r,c,i] == 1;

# each row has each color exactly once
subto column :
   forall <r> in Rows:
   forall <i> in Colors :
      sum <c> in Columns : z[r,c,i] == 1;

# each column has each color exactly once
subto row :
   forall <i> in Colors:
   forall <c> in Columns :
      sum <r> in Rows : z[r,c,i] == 1;

# each height has each color exactly once
subto height :
   forall <h> in Heights :
   forall <i> in Colors :
      sum  <r,c> in Rows * Columns with heights[r,c] == h :
         z[r,c,i] == 1;
```

ZIMPL Model (Objective function)

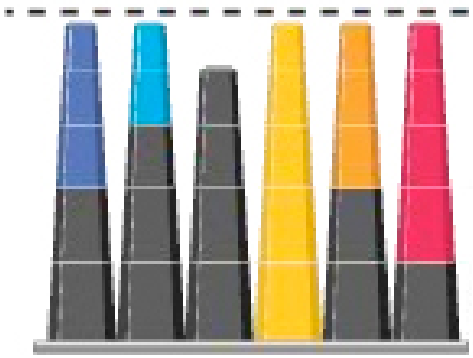
```
# try to find as many feasible positions as possible
maximize obj :
      sum <r,c,i> in Rows * Columns * Colors : z[r,c,i];

# each position gets one color
subto color :
      ...
      sum <i> in Colors : z[r,c,i] <= 1;

# each row has each color exactly once
subto column :
      ...
      sum <c> in Columns : z[r,c,i] <= 1;

# each column has each color exactly once
subto row :
      ...
      sum <r> in Rows : z[r,c,i] <= 1;

# each height has each color exactly once
subto height :
      ...
      sum  <r,c> in Rows * Columns with heights[r,c] == h :
         z[r,c,i] <= 1;
```

A solution for the 36 cube

36 Cube



Place one of each colour in every row and column.



Towers must fit to form a level cube.

36 Towers
Genius!

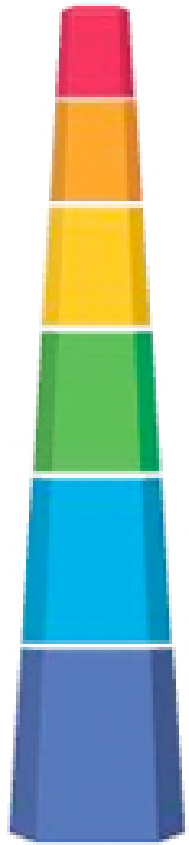
35 Towers
Liar!

31-34 Towers
Rockstar!

26-30 Towers
Contender

20-25 Towers
Varsity

19 or Fewer Towers
Novice



A solution for the 36 cube

Questions

How many solutions exist for the 36 cube?

Mathematics of Infrastructure Planning

The thirty-six officers problem

Stefan Heinz

Zuse Institute Berlin (ZIB)

DFG Research Center MATHEON
Mathematics for key technologies

19/April/2012