

6-Snakes Problem (snake)

— [Run LPL Code](#) , [HTML Document](#) —

Problem: The 6-snakes problem is a board game for one person. The board is a hexagon with 72 notches where the 72 pellets must be placed (see Figure 1¹). The pellets are divided into 6 groups of 12 pellets. In each group the pellets have the same color – hence there are 6 colors.

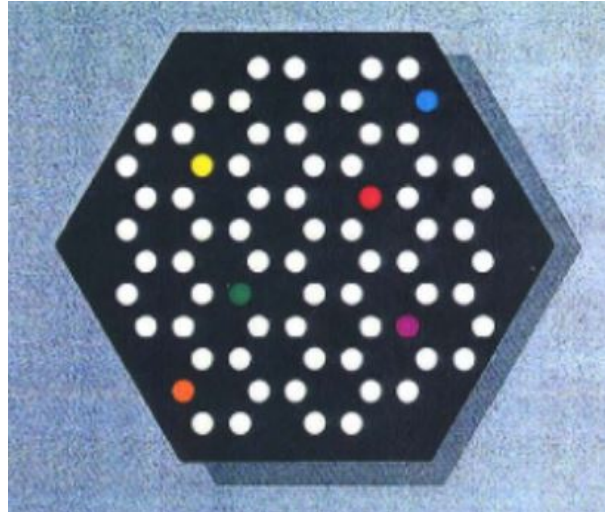


Figure 1: The Game Board

In each group, exactly one pellet is a little bit larger that is called the “head of the snake”. The initial situation of the game is to place the 6 heads of the snakes anywhere in a notch of the board. The goal is to place all other pellets such that the pellets with the same color form a “snake” on the board starting with the head of the snake, that is, each following pellets with the same color must be “neighbor” of the previous pellet with the same color and the snake cannot cross over another snake. A partial solution is given by Figure 2². The game is also called “Glastropfenspiel” or “Schlangengrube” in German.



Figure 2: An almost finished Solution

¹copied from <https://www.spielregeln-spielanleitungen.de/spiel/das-glastropfenspiel/>

²found on <http://www.pyramo.de/schlangengrube.php>

Modeling Steps

The set of notches (or pellets (72 in total)) is defined as a set $I = \{1, \dots, 72\}$ with the indexes $i, j \in I$. There are 6 groups (6 snakes) also introduced as a set $H = \{1, \dots, 6\}$ with index $h \in H$.

First, the hexagonal board is mapped to a graphic and the x - and y -coordinates of all 72 notches are calculated. For this purpose, a grid is defined with the $(0, 0)$ coordinate at the top left (see Figure 3, left side). There are 14 vertical and 12 horizontal lines in the grid. The vertical distances between the horizontal grid lines are fixed at 1 unit. Then the x -coordinates of the vertical grid lines are calculated as follows (to get the nice hexagonal arrangement of the notches inside the board):

$$X_1 = 0 \quad , \quad X_p = X_{p-1} + \begin{cases} 2/\sqrt{3} & , \text{ if } p \bmod 2 = 1 \\ 1/\sqrt{3} & , \text{ if } p \bmod 2 = 0 \end{cases} \quad \text{for all } p \in \{1, \dots, 14\}$$

Two notches $i \in I$ and $j \in I$ are defined to be “neighbors” if the Euclidean distance between the two notches is $\leq 2/\sqrt{3}$. If two notches are neighbors then a line is drawn to link them (see Figure 3, right side).

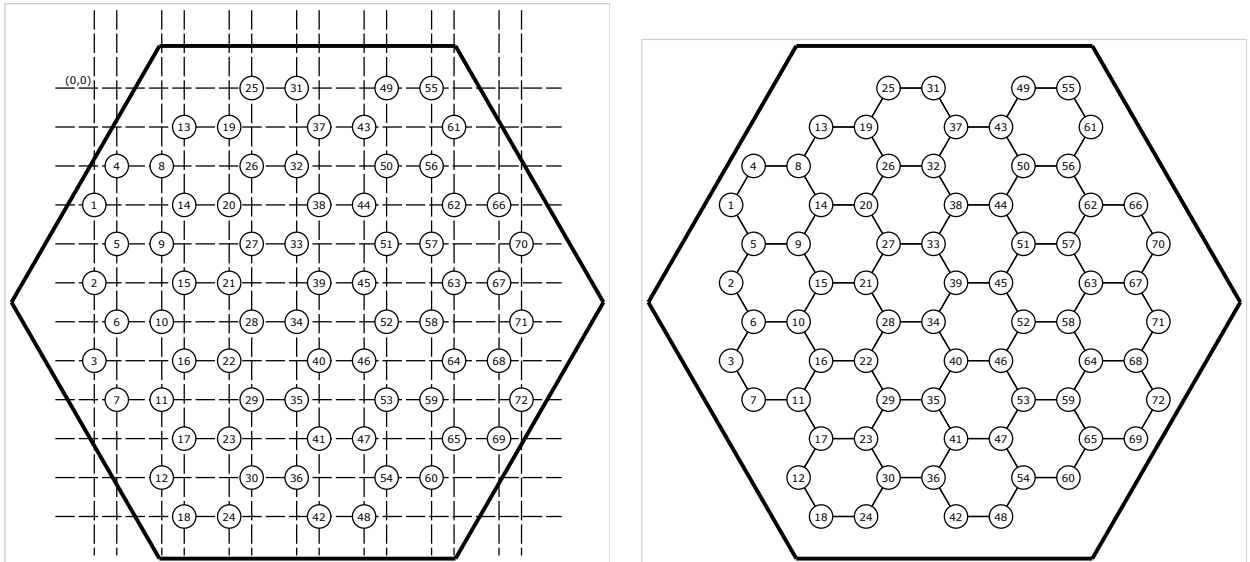


Figure 3: Mapping the Game Board to a $x - y$ Graphic

To start the game, the 6 heads of the snakes (the six larger pellets) are placed anywhere on the board, for example at the notches $(1, 29, 34, 39, 44, 50)$ as in Figure 4. A possible snake is also drawn in the same Figure. It consists of the head at notch 1, it continues to the neighbor notch 4, then to 8 and to 13, and so on, until the tail notch at 61. Hence, a “snake” is a *path* of length 11 (consisting of 12 pellets) starting at the head “jumping” only from neighbor to neighbor.

“The path distance of a notch from the head plus 1” is defined as *position*. Hence, the head has position 1 and its neighbor notch 4 has position 2, and the tail notch 61 has position 12. Let’s introduce an (ordered) set $K = \{1, \dots, 12\}$ with index $k \in K$ to denote the position of a notch inside a snake.

Now finally, we can introduce a binary variable $z_{i,h,k}$ with the following meaning: $z_{i,h,k} = 1$ if and only if notch i is in snake h at the position k (otherwise $z_{i,h,k} = 0$).

Using this definition of the variables the model constraints are very straightforward:

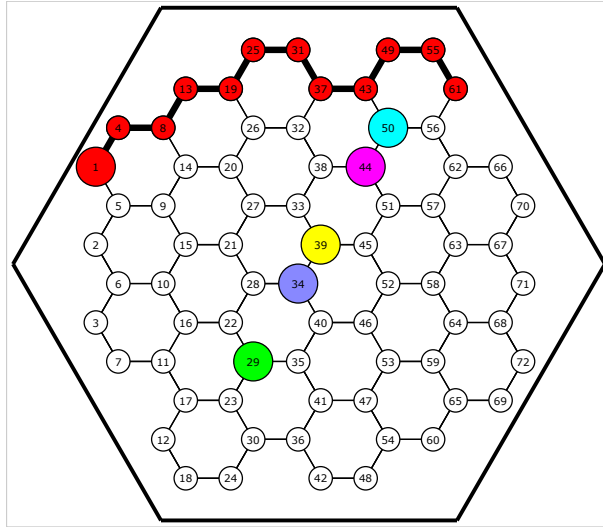


Figure 4: An complete snake and all 6 Heads

1. Let a subset $J \subset I$ with $|J| = 6$ be the head notches, then the six heads have position 1, that means:

$$z_{i,h,1} = 1 \quad , \text{ for all } i \in J, h \in H$$

2. Each notch is exactly in one snake and has a unique position:

$$\sum_{h,k} z_{i,h,k} = 1 \quad , \text{ for all } i \in I$$

3. The same snake and the same position is assigned to exactly one notch:

$$\sum_i z_{i,h,k} = 1 \quad , \text{ for all } h \in H, k \in K$$

4. Finally, this is the key constraint: If a notch in a snake has position k then there must be at least one other neighbor that has position $k + 1$ (except the tail notch). One can formulate this as a logical constraint as follows (where $e_{i,j}$ is a relation that is true if i and j are neighbors and $K^- = K \setminus \{12\}$)

$$z_{i,h,k} \rightarrow \bigvee_{j \in I | e_{i,j}} z_{i,h,k+1} \quad \text{for all } i \in I, h \in H, k \in K^-$$

LPL translates this constraint automatically into a linear form as follows (which can be alternatively formulated).

$$\sum_{j \in I | e_{i,j}} z_{i,h,k+1} \leq z_{i,h,k} \quad \text{for all } i \in I, h \in H, k \in K^-$$

5. We are looking only for a feasible solution, if one exists.

Listing 1: The Main Model implemented in LPL [2]

```

model snake "6-Snakes Problem";
set i, j "set of notches";

```

```

e{i,j} "i and j are neighbours";
h      "set of snakes";
k      "Length of snakes (ordered set of position)";
parameter J{h} "6 starting positions (heads of snake)";
binary variable z{i,h,k};
constraint
A{h}: z[J,h,1] = 1;
B{i}: sum{h,k} z = 1;
C{h,k}: sum{i} z = 1;
--D{i,h,k|k<#k}: sum{j in e} z[j,h,k+1] >= z[i,h,k]; //-SL(1,1000);
D{i,h,k|k<#k}: z[i,h,k] -> or{j in e} z[j,h,k+1];
solve;
end

```

Solution: A solution for the initial placements of the head as given above is given in Figure 5

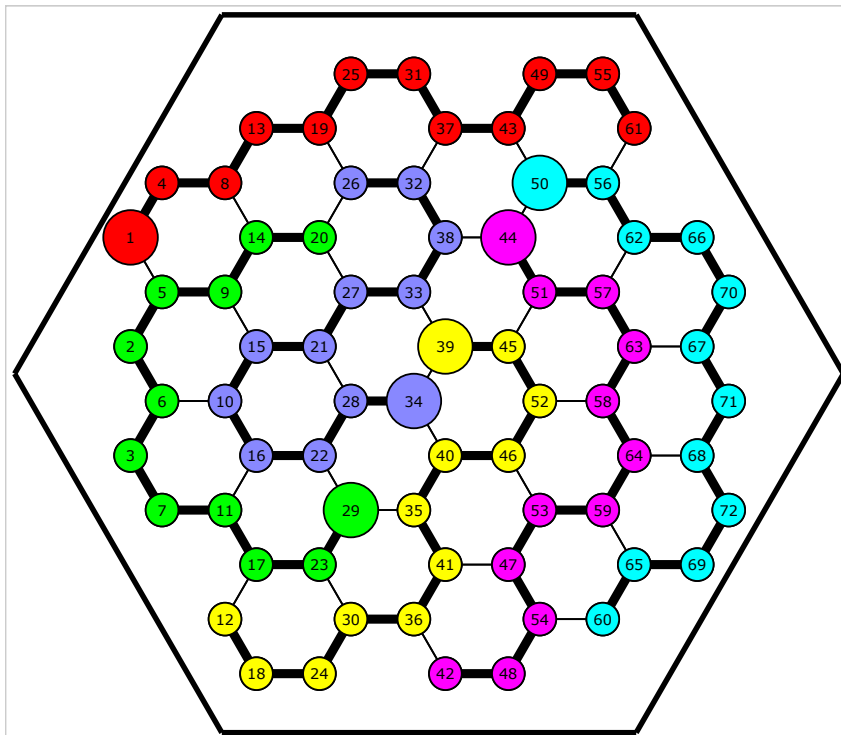


Figure 5: Snake: a solution

Gurobi 9.0 takes 6secs to find this solution. The problem is not only very compact but also efficient. This underlines once again the importance to choose the “right” variable. The key in this variable is to integrate the position k inside the snake to get a connected path as is formulated in the last constraint.

Questions

1. For some initial placements of the head, there is no solution. Find one.
2. “Solve” the problem with the subset J in the previous answer.

Answers

1. A subset is $J = \{1, 2, 3, 70, 71, 72\}$. The solver returns an infeasibility. One can relax the last constraint using the LPL `SL()` function to find the best possible solution cutting one snake into two pieces.

2. The problem is infeasible, so no solution exists. However we can find a solution with a minimal “deviation” from a feasible solution. Replace the constraint

$$| \quad D\{i, h, k | k < \#k\}: \text{sum}\{j \text{ in } e\} z[j, h, k+1] \geq z[i, h, k]$$

with the constraint

$$| \quad D\{i, h, k | k < \#k\}: \text{sum}\{j \text{ in } e\} z[j, h, k+1] \geq z[i, h, k] - S1(1, 1000);$$

and solve it. The result is shown in Figure 6 (one snake is cut into two pieces).

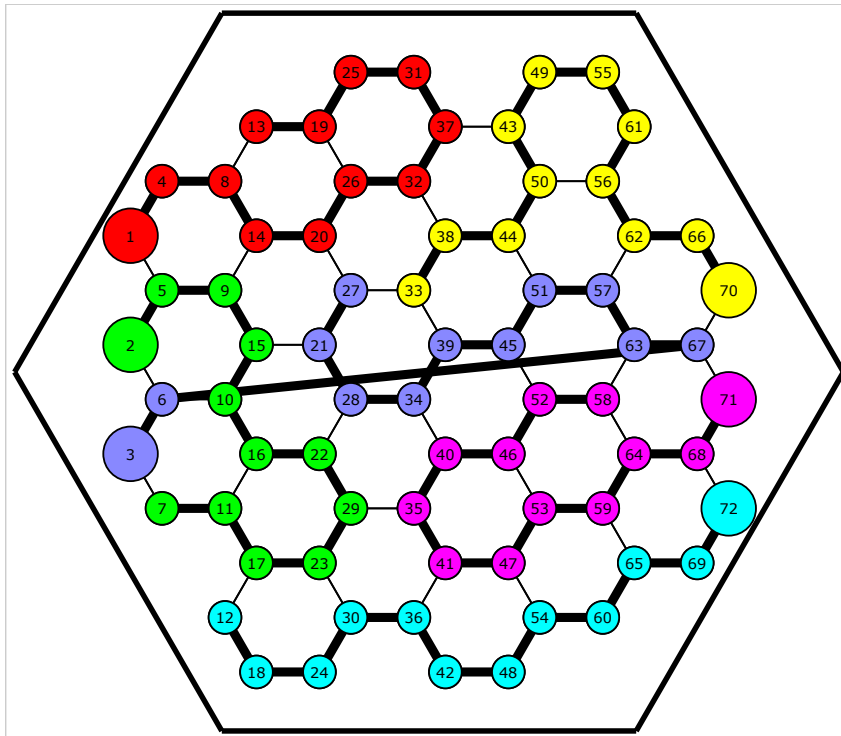


Figure 6: A solution minimizing the violations

References

- [1] MatMod. Homepage for Learning Mathematical Modeling : <https://matmod.ch>.
- [2] Hürlimann T. Reference Manual for the LPL Modeling Language, most recent version. <https://matmod.ch/lpl/doc/manual.pdf>.