Yashi game

SAT ENCODING

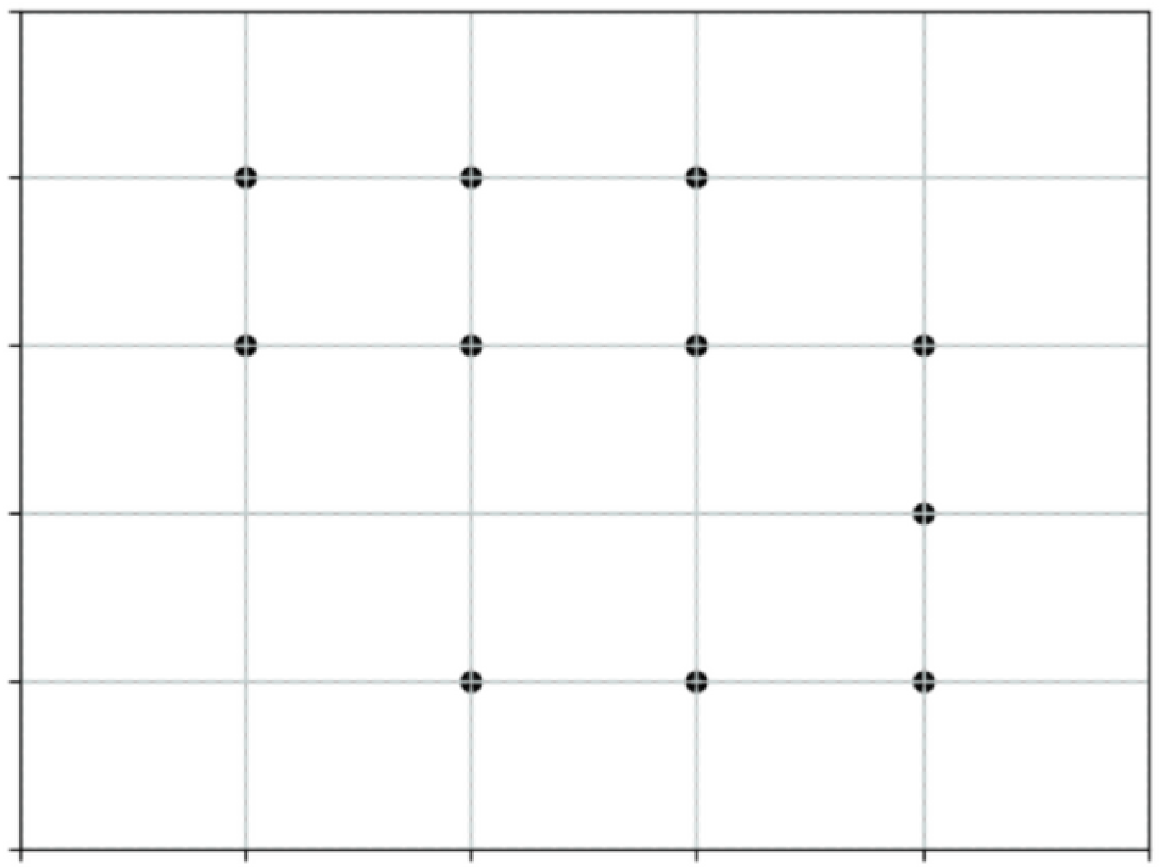
By Camilla Colanero

Goal

We aim to find a minimum-length solution of a configuration G of the Yashi game

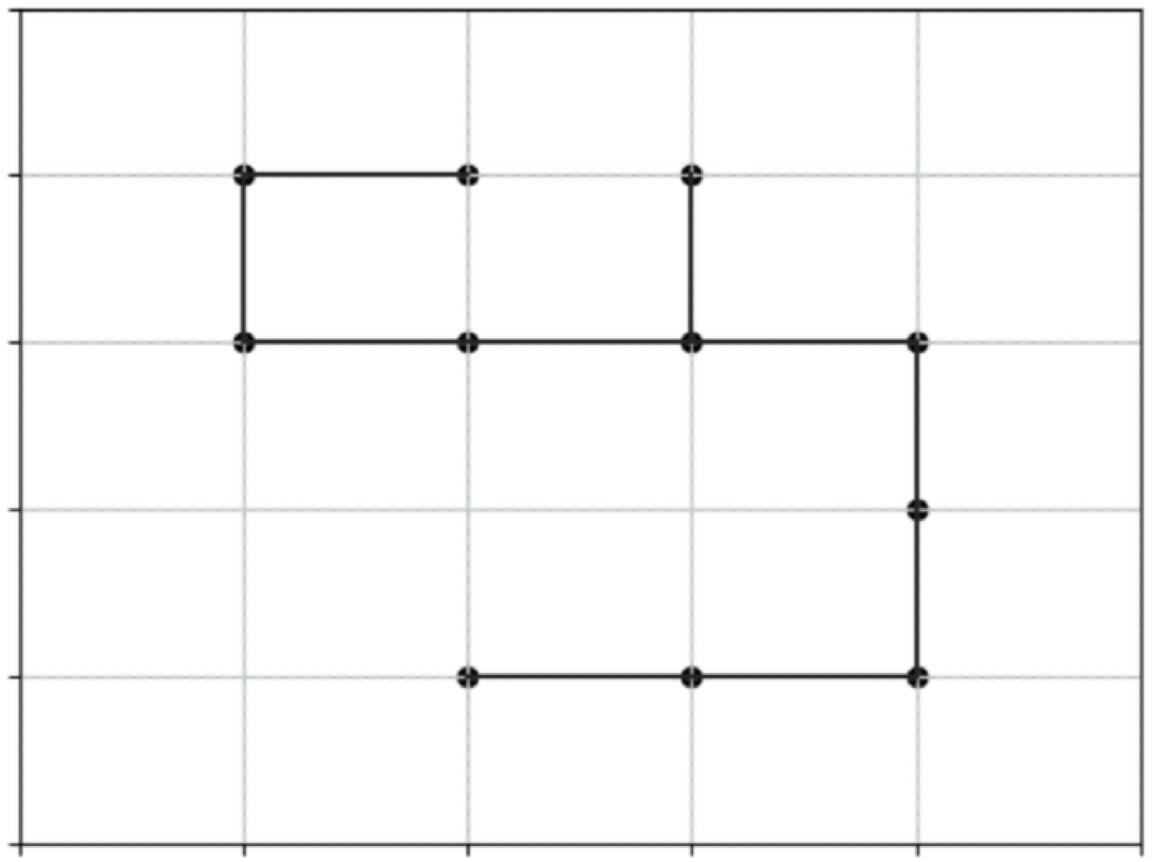
Yashi Game

INSTRUCTIONS

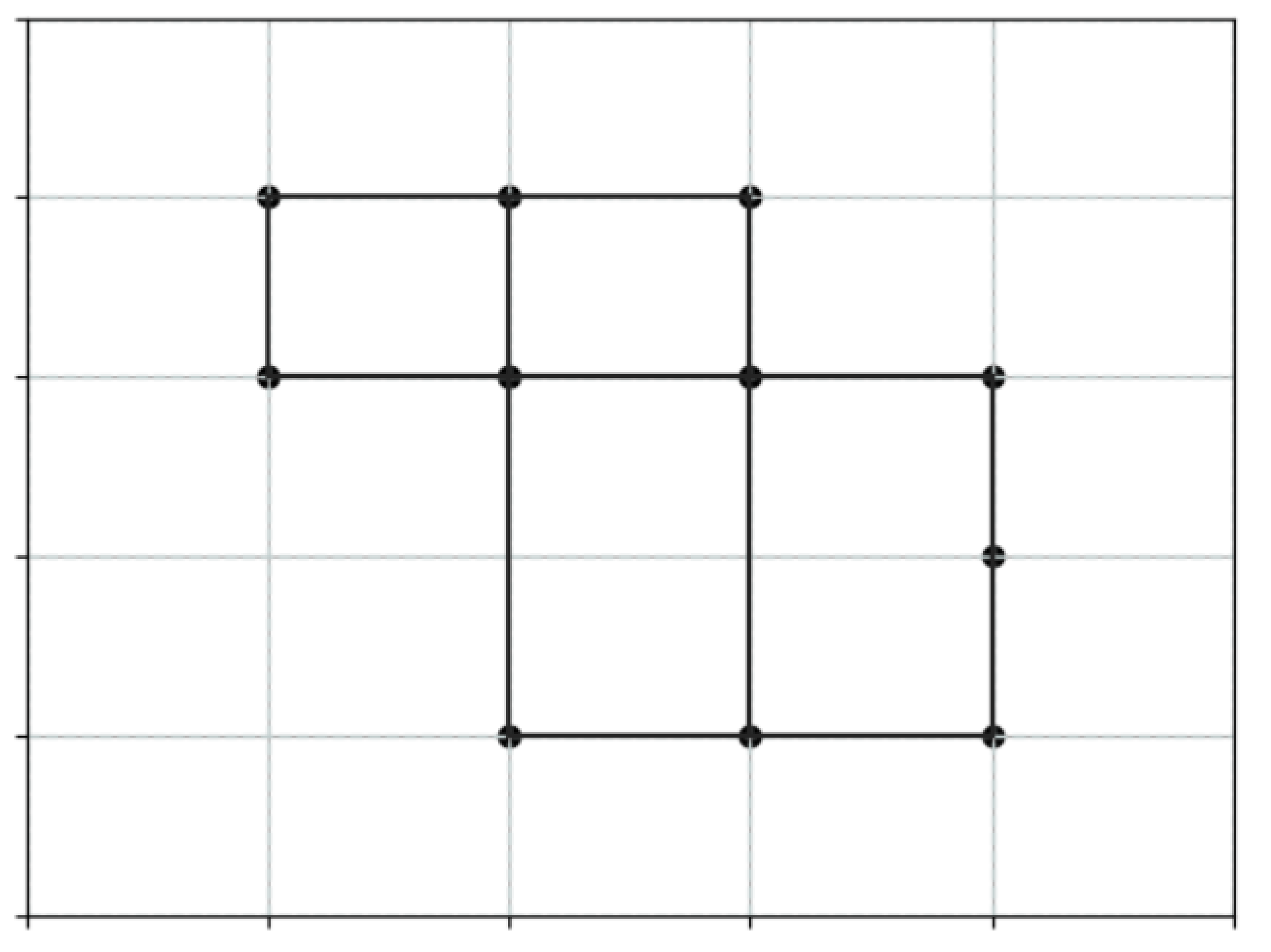


A Yashi game consists in trying to connect all the dots with only vertical or horizontal segments with this conditions:

1. Two segments cannot cross each other
2. There cannot be cycles.



variables CREATION



We assign to each link that can be drawn a unique index.

In this way, in the final solution we have:

$$\begin{cases} i, & \text{if the } \textit{i-th} \text{ link is present} \\ -i, & \text{if the } \textit{i-th} \text{ link is not present} \end{cases}$$

hard CLAUSES

This are the clauses that must be satisfied in order to have a solution of the Yashi Game. If even one of them is never satisfied, there is no solution.

1. **No crossing links**
2. **No cycles**
3. **No isolation**
4. **Exactly k links**

no

CROSSING LINKS

Two links cannot cross each other. To avoid this we create clauses for each couple of links that cross each other.

If the *i-th* link and the *j-th link cross each other, we will have the clause*:

$$\neg i \lor \neg j$$

no CYCLES

We look for all the cycles that pop up when all the links are present.
For each set C of links that create a cycle, we create a clause such that all the links inside of C cannot be present together.
So, for each C, the clause will be:

$$\bigvee_{i \in C} \neg i$$

no ISOLATION

We cannot have a point that is not linked to any other point.
Therefore we impose the clause that at least one link for each point is present.
That is, we create the clauses, for each S_x that is the set of all the links that start from the point *x:*

$$\bigvee_{i \in S_x} i$$

If the point x cannot be linked to any other point, we add the clause that is unsatisfiable:

$$\neg 1 \vee 1$$

exactly k LINKS

We want exactly k links in the solution, where k is the number of points - 1 .
The constraint exactly k propositional variables in X are true is formalized as:

$$\bigwedge_{\substack{I\subseteq[n]\\|I|=n-k+1}} \bigvee_{i\in I} x_i \wedge \bigwedge_{\substack{I\subseteq[n]\\|I|=k+1}} \bigvee_{i\in I} \neg x_i$$

In our case n is the number of possible links.
Therefore, we add all the clauses relative to at most k links and at least k links.

soft CLAUSES

This are the clauses where a weight is assigned to the truth value of the clause. With the MaxSAT we want to minimize the weight of the clauses that are assigned to false.

We assign to the absence of each link a weight, that corresponds to minus its length.
Therefore if the solver has to choose a link between two links, it will choose to mantain the link with smaller length in order to have the minimum weight.

For example, if we have the clauses:
[1 , weight = -2] , [4 , weight = -4]
and we have to satisfy the clause $-1 \lor -4$ we will have in the solution the link 1.

demo

THANKS FOR YOUR ATTENTION