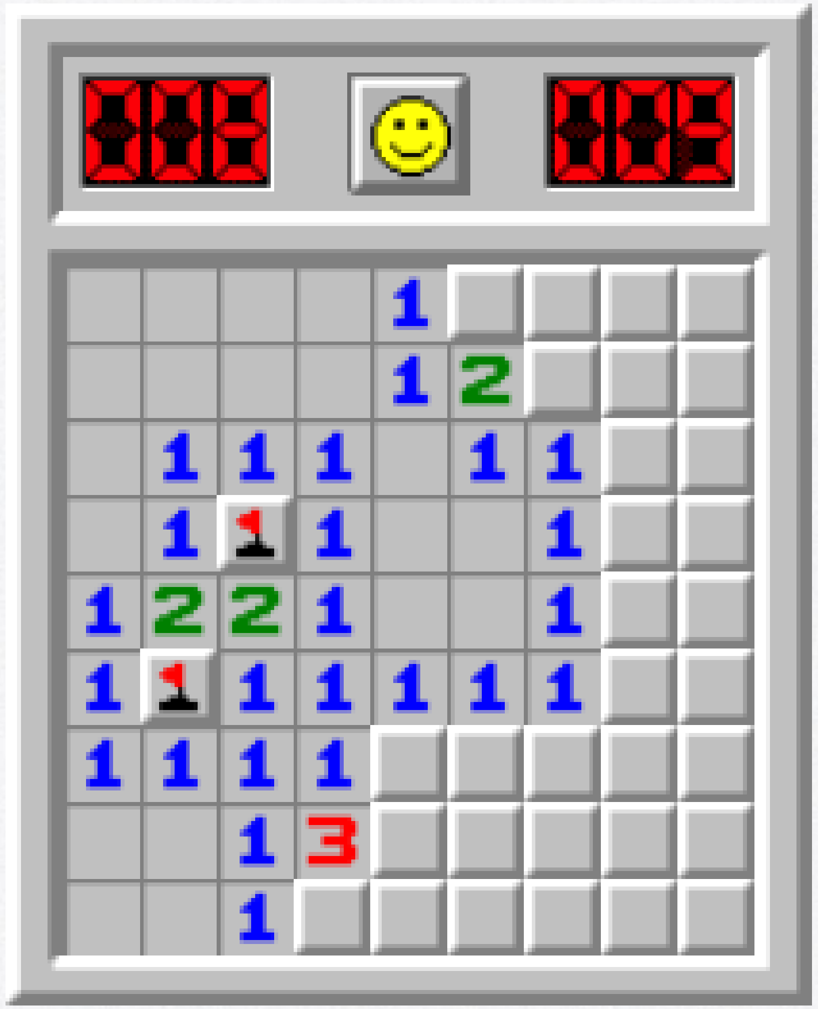
The Bullets Puzzle: a Paper-and-Pencil Minesweeper

Todd W. Neller, Hien Tran
Gettysburg College

What is Minesweeper?

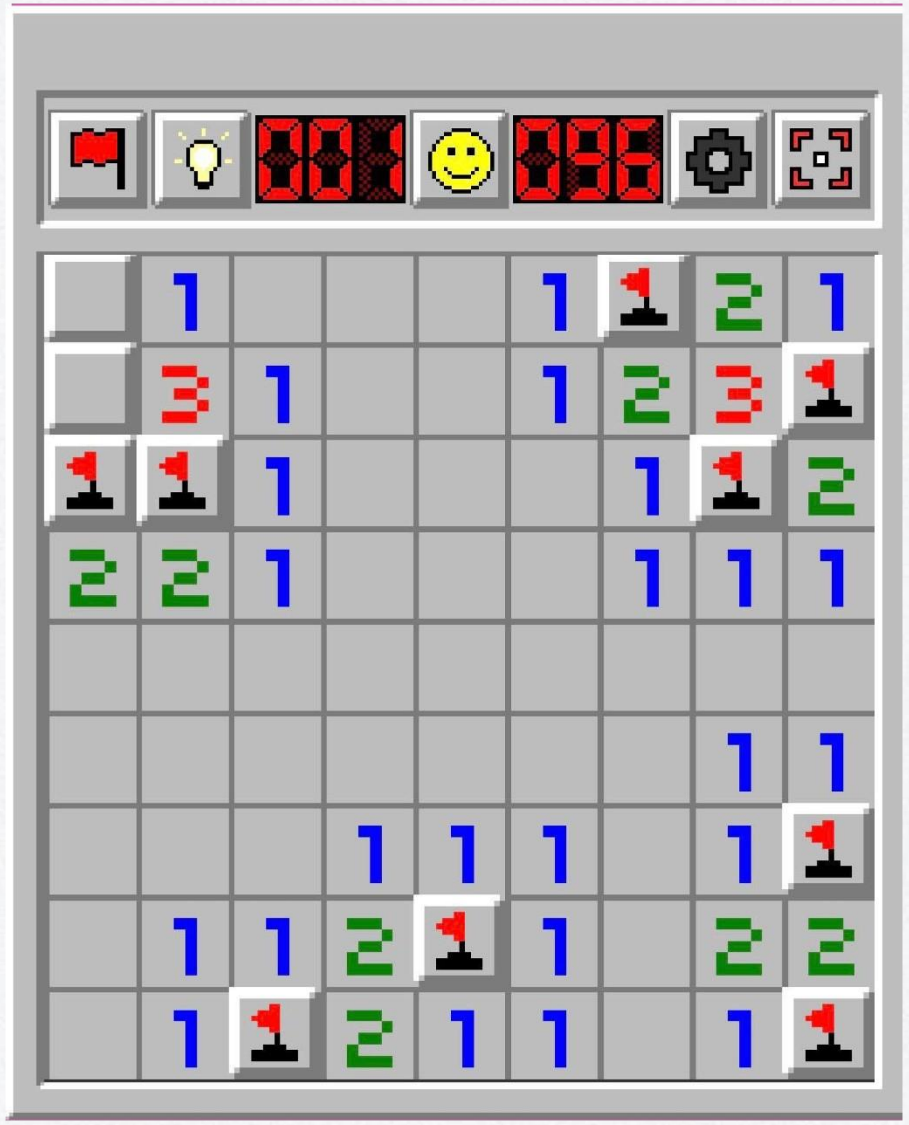
- A single-player game which objective is to clear a rectangular board containing hidden mines.
- Players must use clues about the number of neighboring mines in each field.
- https://minesweeper.online/game/1136538424



What’s Wrong with Minesweeper?

- No initial information.
- If a mine is revealed, the player loses. Thus, the first click may reveal clue information dynamically, or end the game with the revelation of a mine.
- Logic puzzles should be solvable through logic alone.
- Need for guessing is a flaw in design → Bad Puzzle

A situation where a guess must be made



Example Bullets Puzzle with 16 bullets and solution:

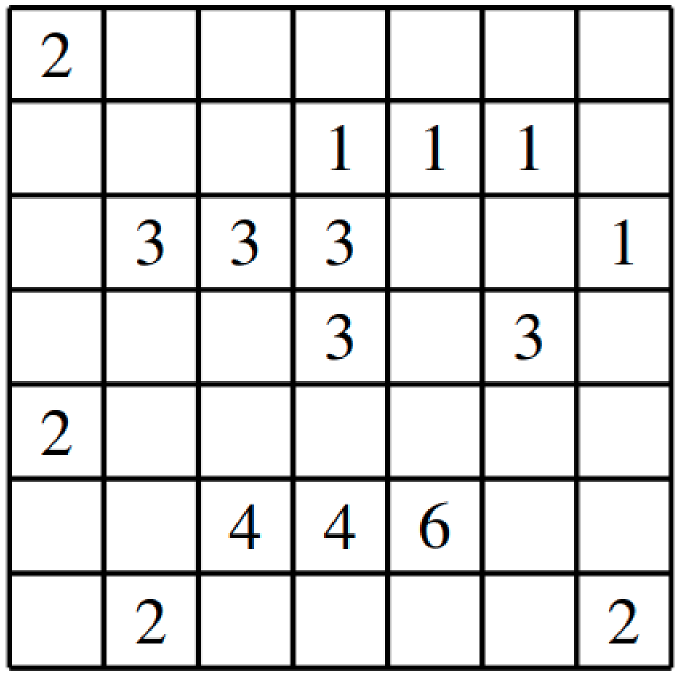


Figure 1: Example Bullets puzzle with 16 bullets

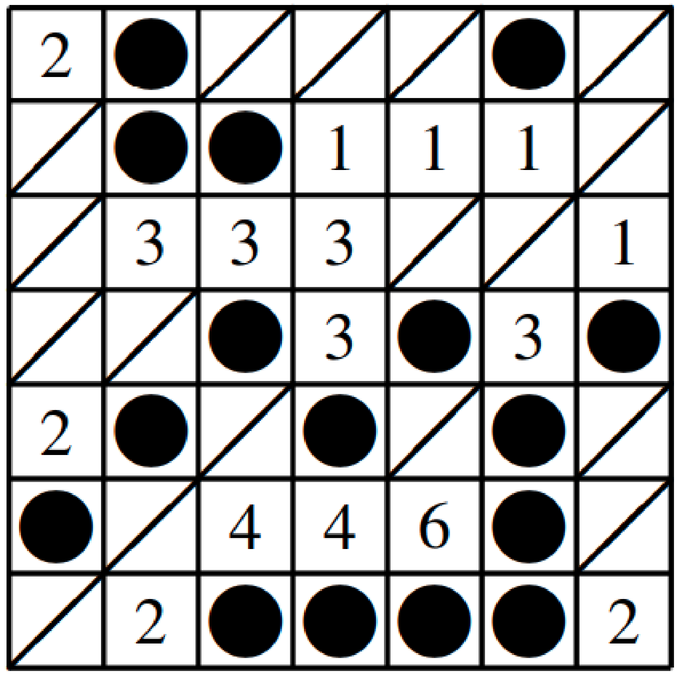
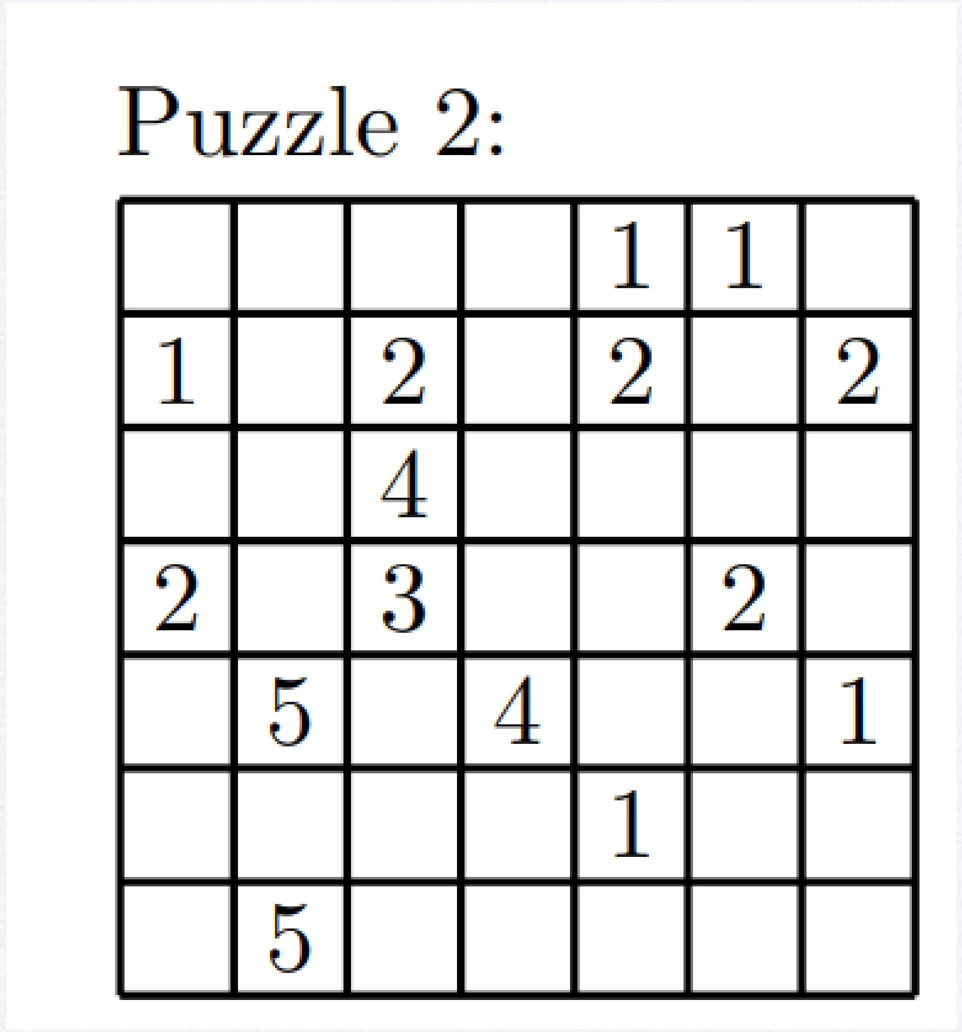


Figure 2: Solution to example Bullets puzzle

About our Research:

- Design and generate the Bullets puzzle, a paper-and-pencil variant of Minesweeper with no guessing necessary.
- Developed a reasoning engine to produce an explanation of a human-like sequence of solution steps.
- Provide insights to subjective puzzle quality, minimal clue sampling trade-offs, and optimal bullet density.

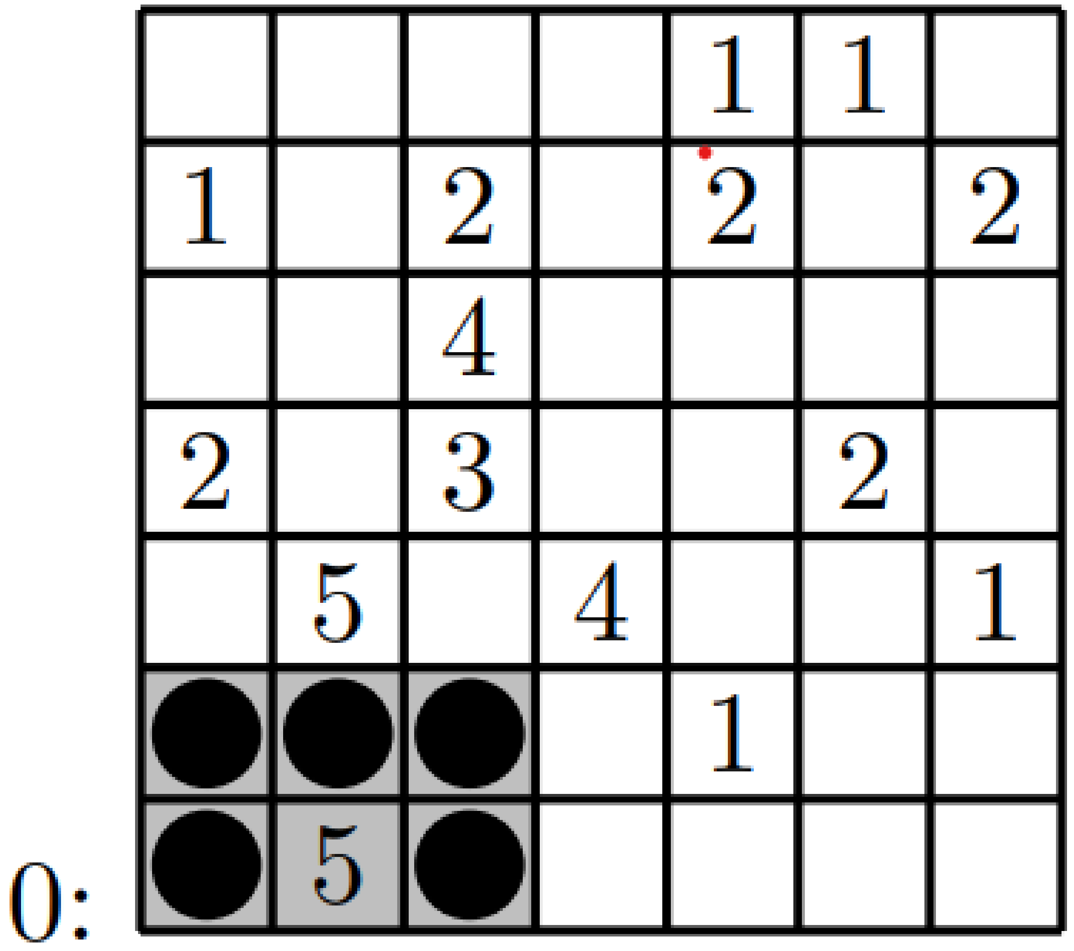
Let’s play a game together



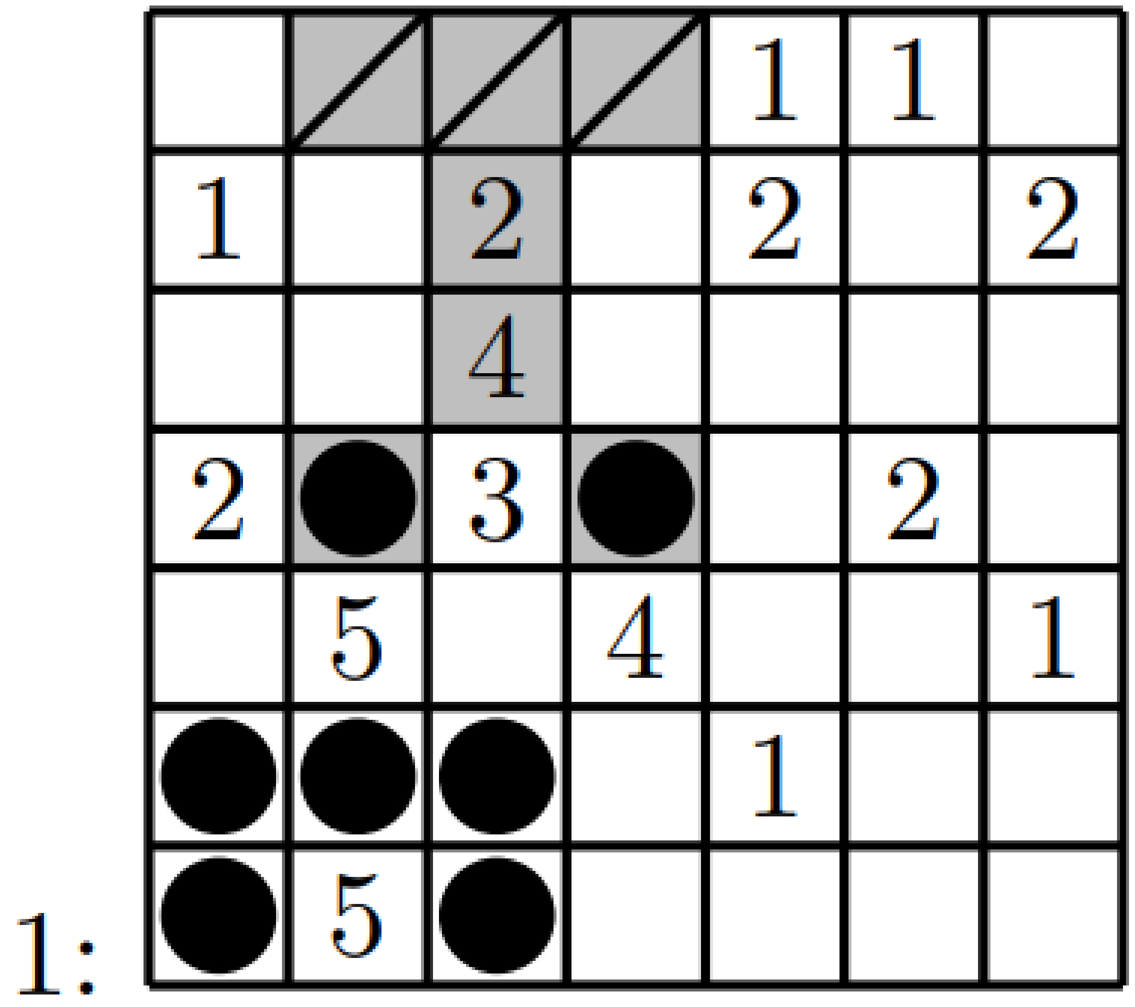
Puzzle 2:

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

Step 1:

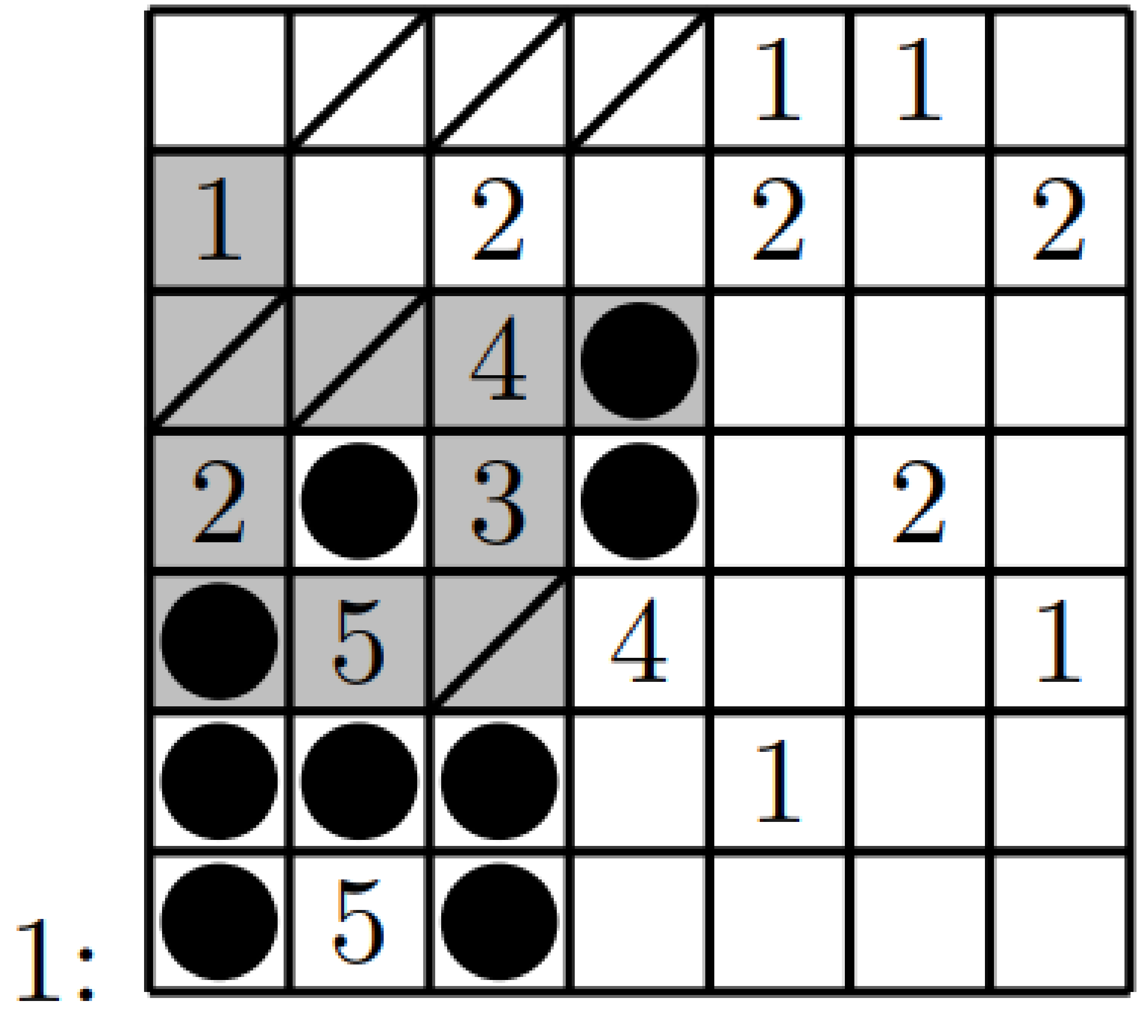


Step 2:



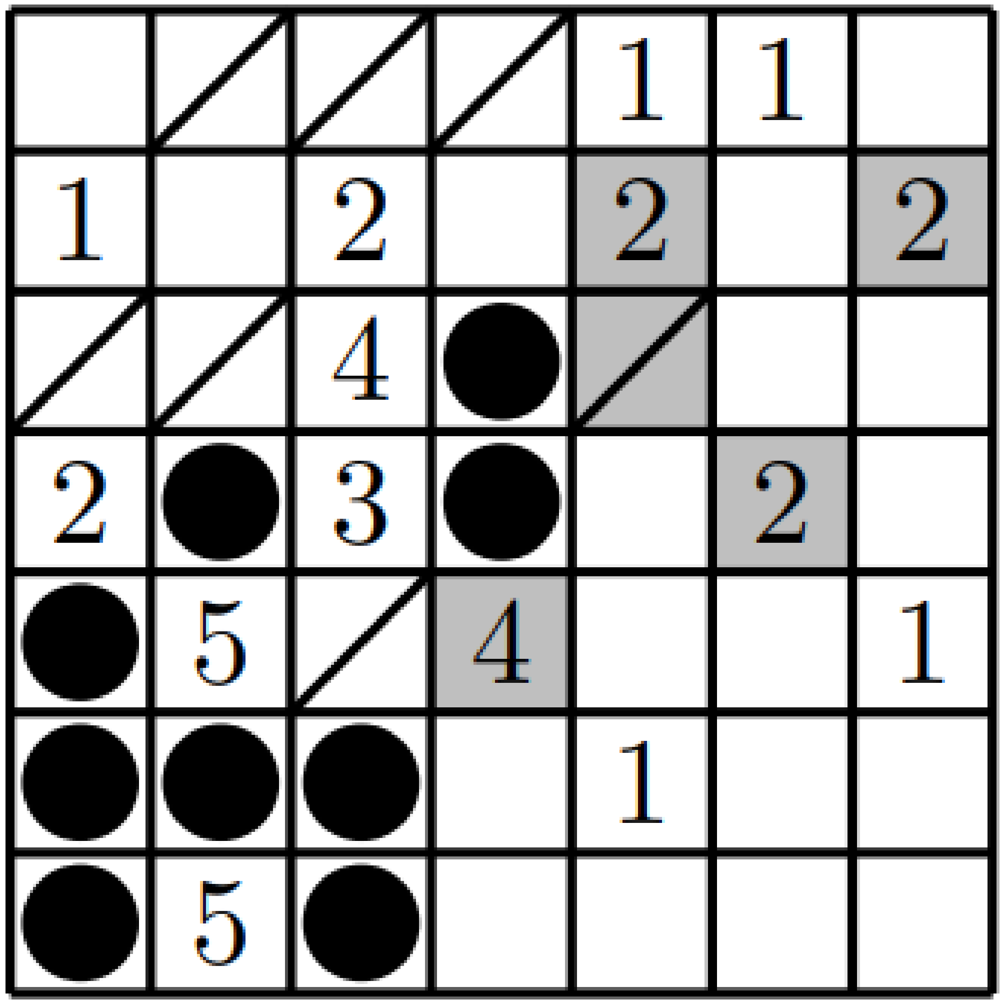
	▨	▨	▨	1	1	
1		2		2		2
		4				
2	●	3	●		2	
	5		4			1
●	●	●		1		
●	5	●				

Step 3:

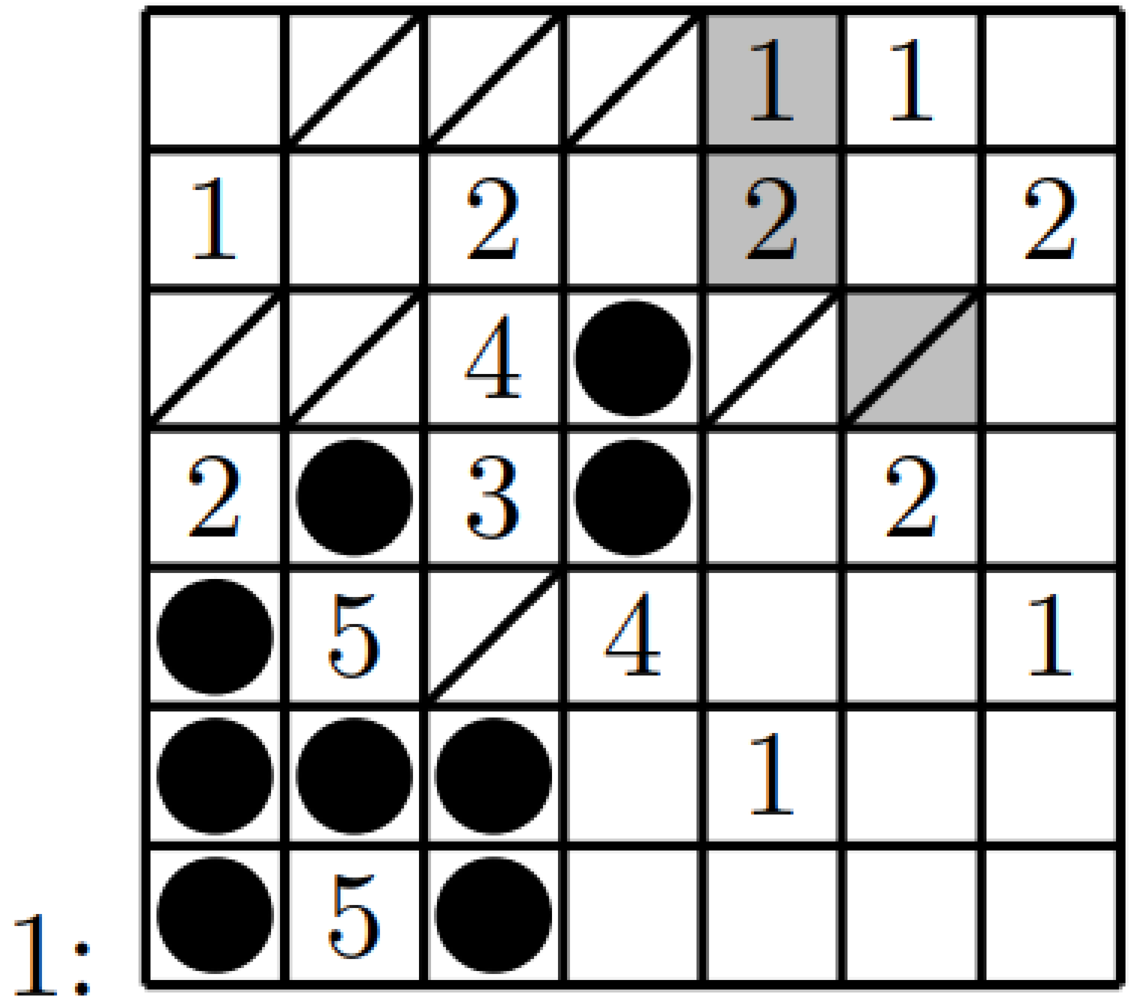


Step 4:

1:

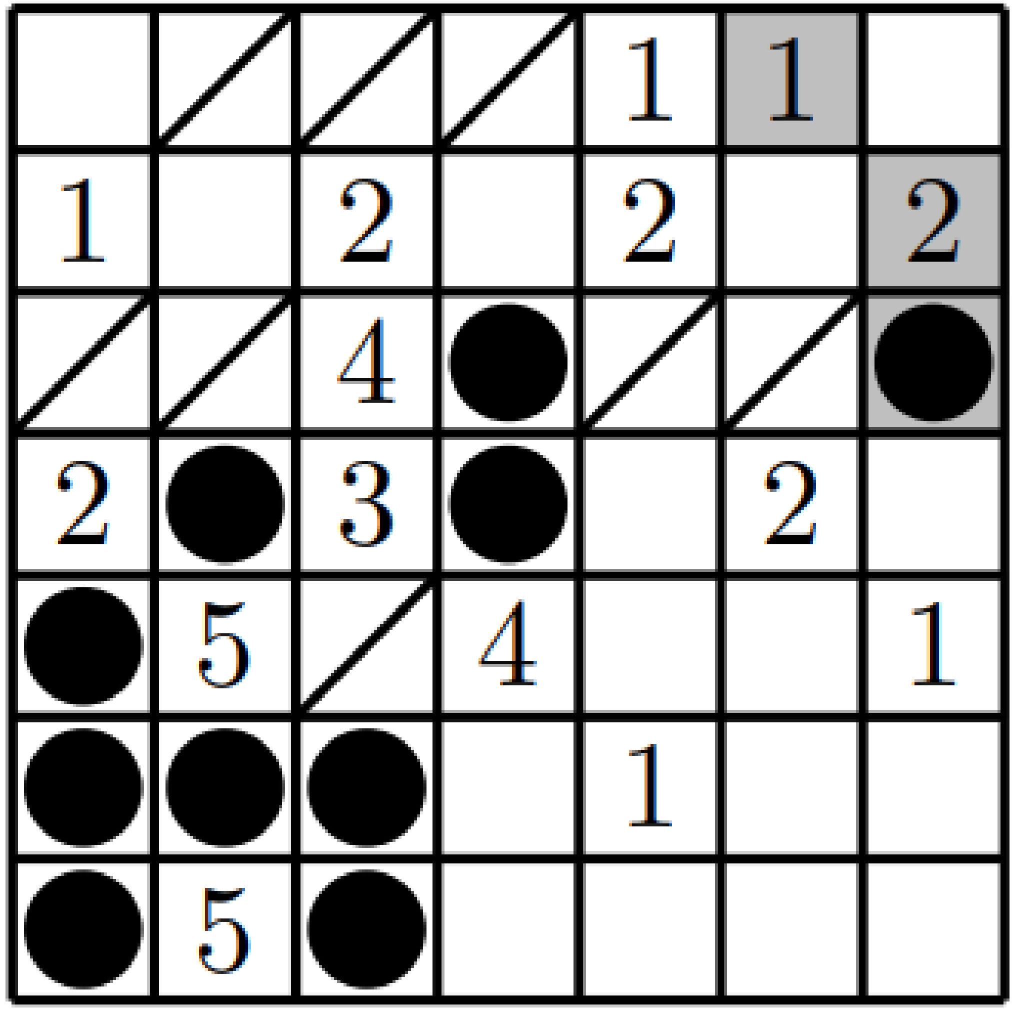


Step 5:

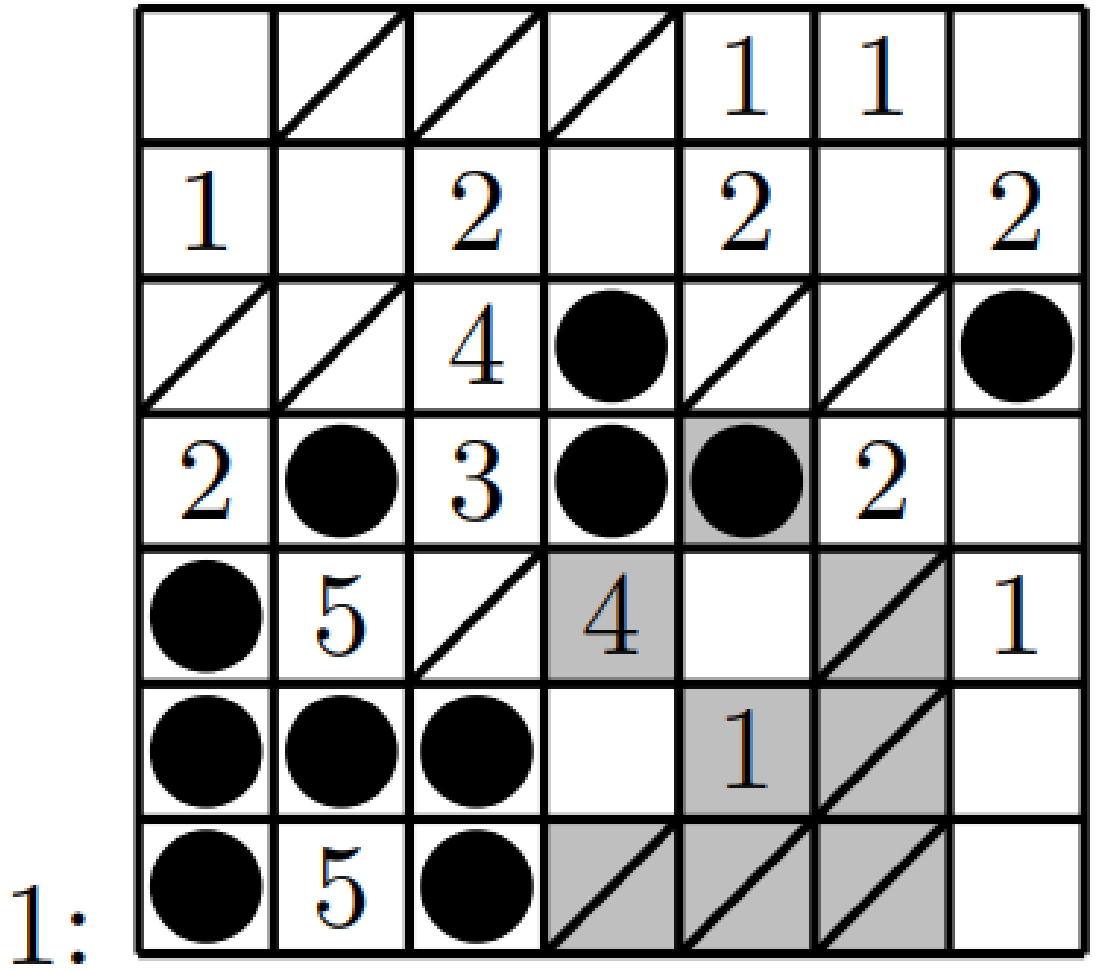


Step 6:

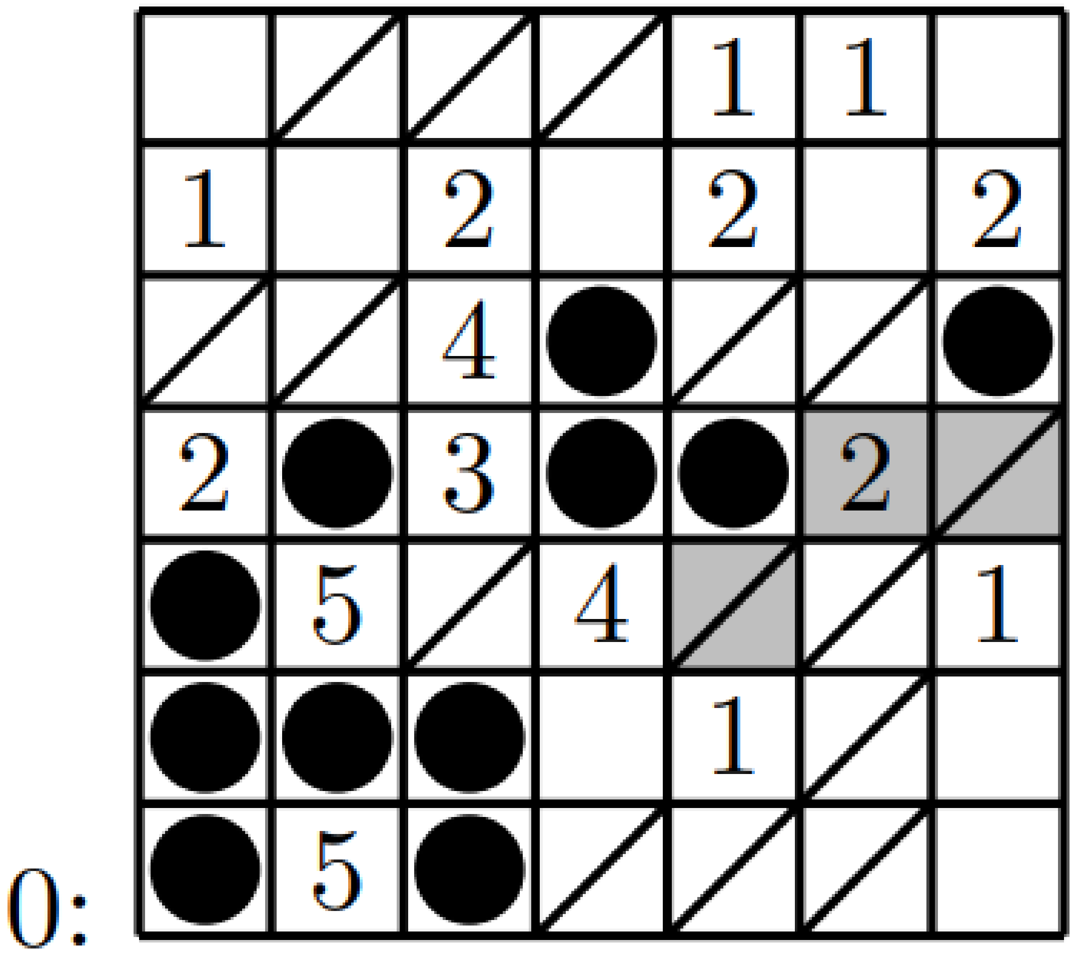
1:



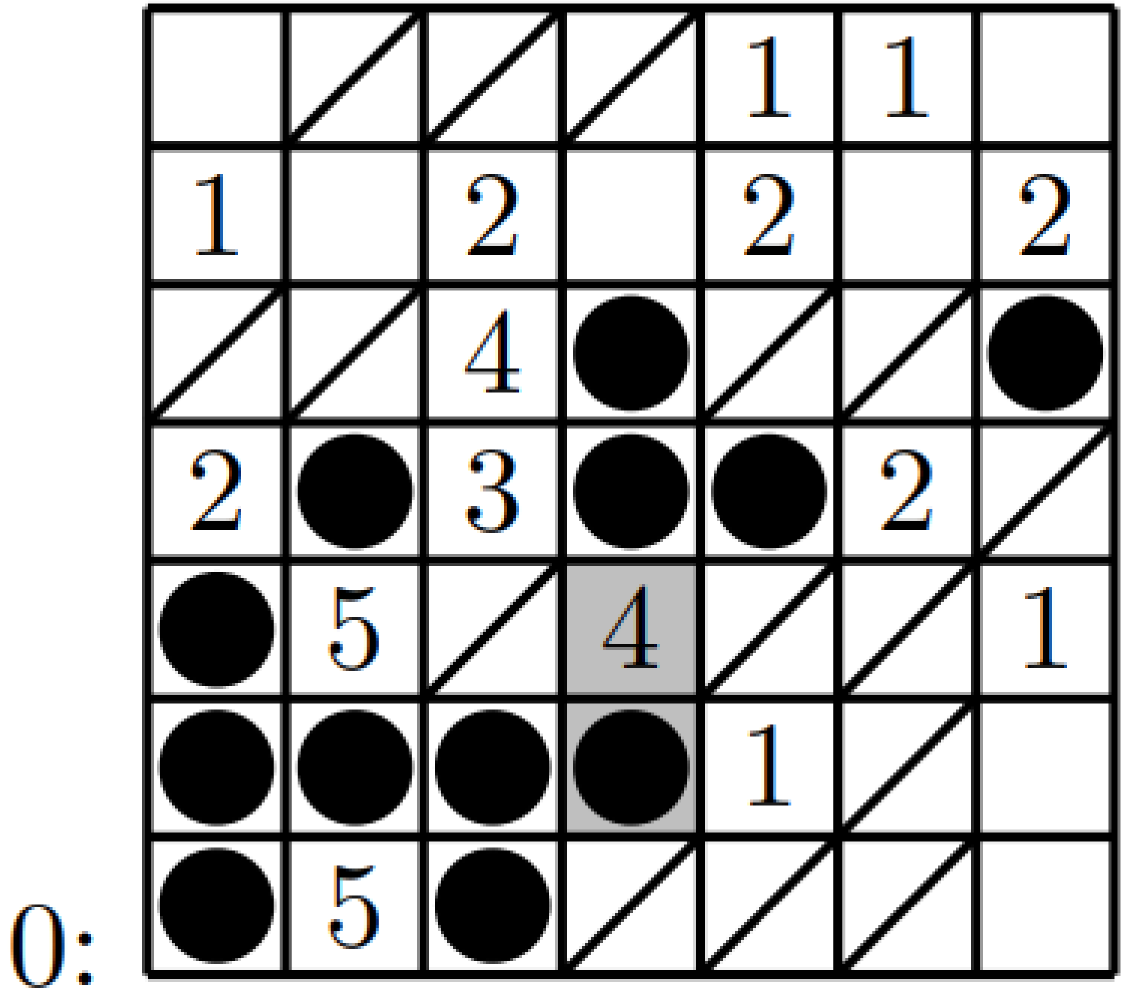
Step 7:



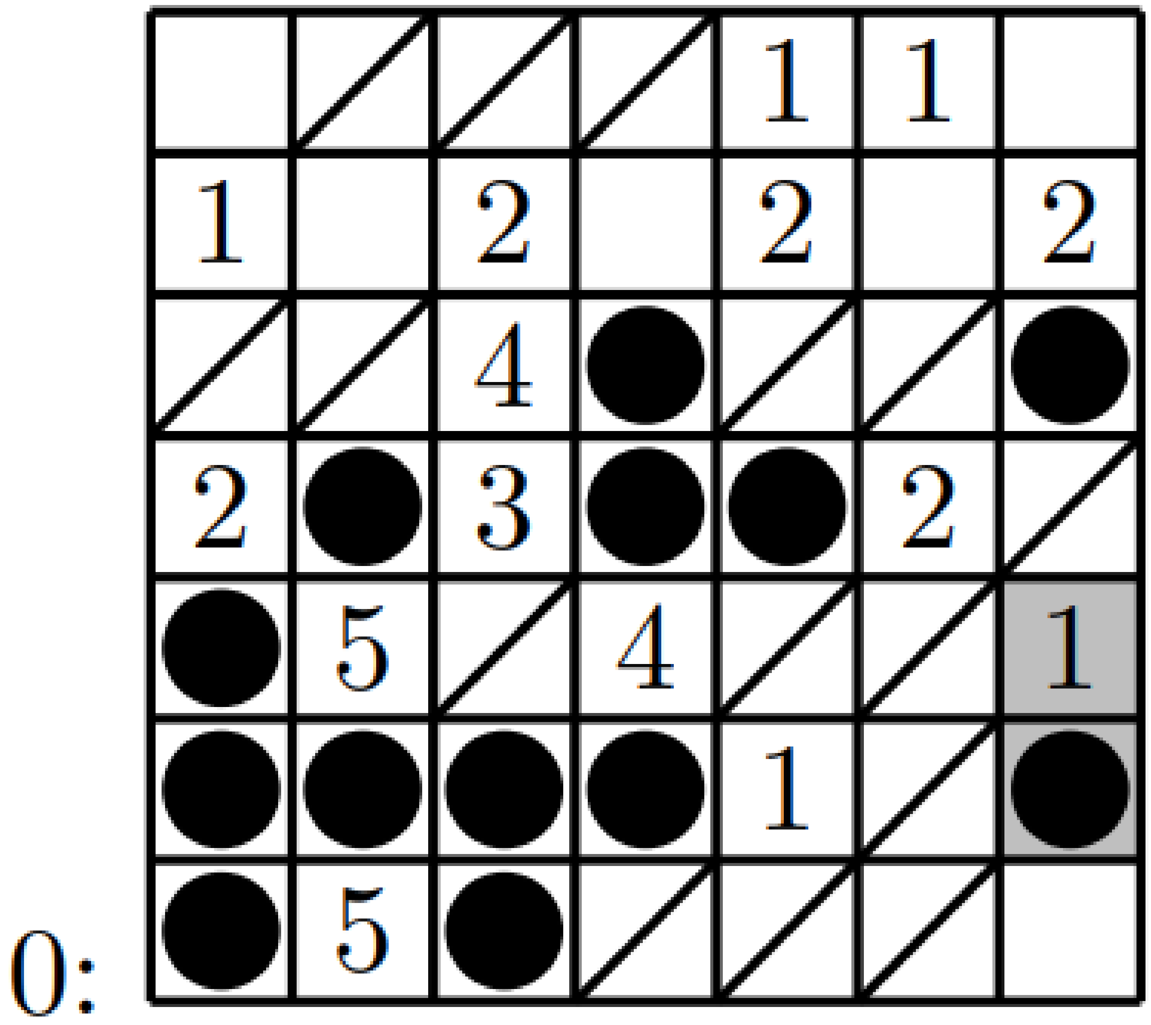
Step 8:



Step 9:

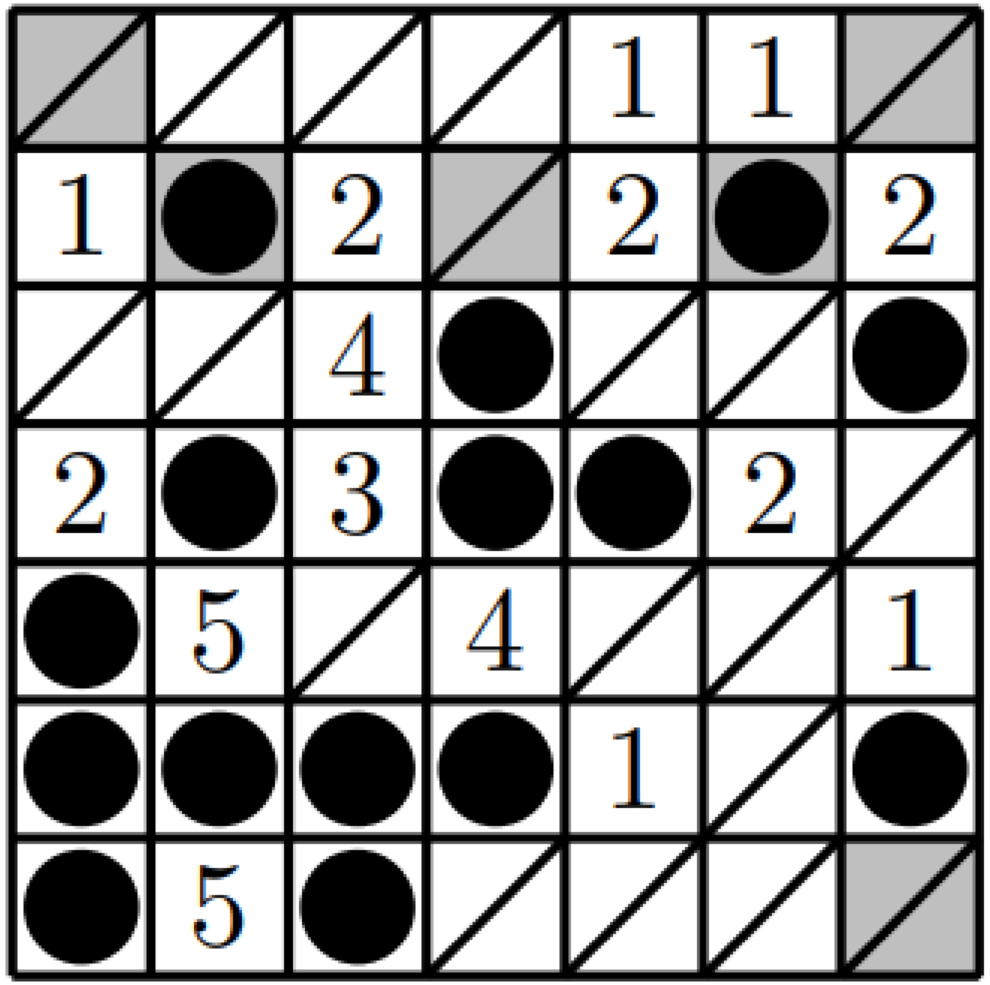


Step 10:



Step 11:

2:



At-Least Constraint Reasoning

Representing Knowledge: Literals

2						
			1	**1**	**1**	
	3	**3**	**3**			**1**
			3		**3**	
2						
		4	**4**	**6**		
	2					**2**

16 bullets

c_0	c_1	c_2	c_3	c_4	c_5	c_6
c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}
c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}	c_{20}
c_{21}	c_{22}	c_{23}	c_{24}	c_{25}	c_{26}	c_{27}
c_{28}	c_{29}	c_{30}	c_{31}	c_{32}	c_{33}	c_{34}
c_{35}	c_{36}	c_{37}	c_{38}	c_{39}	c_{40}	c_{41}
c_{42}	c_{43}	c_{44}	c_{45}	c_{46}	c_{47}	c_{48}

$c_i \in$ {true, false} whether or not a bullet is in cell i

Representing Knowledge: Easy Clauses

2	c_1					
c_7	c_8					

Exactly 2 bullets are adjacent to c_0:

$(c_1 \wedge c_7 \wedge \neg c_8) \vee (c_1 \wedge \neg c_7 \wedge c_8) \vee (\neg c_1 \wedge c_7 \wedge c_8)$

One disjunction of three conjunctions (clauses)

Representing Knowledge: Not-So-Easy Clauses

2	c_1					
c_7	c_8					
	c_{29}	c_{30}	c_{31}			
	c_{36}	**4**	c_{38}			
	c_{43}	c_{44}	c_{45}			

Exactly 2 bullets are adjacent to c_0:

$(c_1 \wedge c_7 \wedge \neg c_8) \vee (c_1 \wedge \neg c_7 \wedge c_8) \vee (\neg c_1 \wedge c_7 \wedge c_8)$

One disjunction of three conjunctions (clauses)

Exactly 4 bullets are adjacent to c_{37}:

$(c_{29} \wedge c_{30} \wedge c_{31} \wedge c_{36} \wedge \neg c_{38} \wedge \neg c_{43} \wedge \neg c_{44} \wedge \neg c_{45}) \vee$

$(c_{29} \wedge c_{30} \wedge c_{31} \wedge \neg c_{36} \wedge c_{38} \wedge \neg c_{43} \wedge \neg c_{44} \wedge \neg c_{45}) \vee$

…

$\vee (\neg c_{29} \wedge \neg c_{30} \wedge \neg c_{31} \wedge \neg c_{36} \wedge c_{38} \wedge c_{43} \wedge c_{44} \wedge c_{45})$

One disjunction of $\binom{8}{4} = 70$ conjunctions (clauses)

Representing Knowledge: Too Many Clauses

2	c_1					
c_7	c_8					
	c_{29}	c_{30}	c_{31}			
	c_{36}	**4**	c_{38}			
	c_{43}	c_{44}	c_{45}			

16 bullets

Exactly 2 bullets are adjacent to c_0:
$(c_1 \wedge c_7 \wedge \neg c_8) \vee (c_1 \wedge \neg c_7 \wedge c_8) \vee (\neg c_1 \wedge c_7 \wedge c_8)$
One disjunction of three conjunctions (clauses)

Exactly 4 bullets are adjacent to c_{37}:
$(c_{29} \wedge c_{30} \wedge c_{31} \wedge c_{36} \wedge \neg c_{38} \wedge \neg c_{43} \wedge \neg c_{44} \wedge \neg c_{45}) \vee$
$(c_{29} \wedge c_{30} \wedge c_{31} \wedge \neg c_{36} \wedge c_{38} \wedge \neg c_{43} \wedge \neg c_{44} \wedge \neg c_{45}) \vee$
…
$\vee (\neg c_{29} \wedge \neg c_{30} \wedge \neg c_{31} \wedge \neg c_{36} \wedge c_{38} \wedge c_{43} \wedge c_{44} \wedge c_{45})$
One disjunction of $\binom{8}{4} = 70$ conjunctions (clauses)

Exactly 16 bullets are in $\{c_0, c_1, \ldots, c_{48}\}$:
One disjunction of $\binom{49}{16} > 3 \times 10^{12}$ conjunctions (clauses)

Representing Knowledge: At-Least Constraints

- An alternative, compact form of representation is in the form of a set of *at-least constraints*:
  - at-least(n, $\{c_{i1,} \ldots, c_{im}\}$, v) means "At least n of $\{c_{i1,} \ldots, c_{im}\}$ must have value v."

Representing Knowledge: Easy Clauses

2	c_1					
c_7	c_8					

Exactly 2 bullets are adjacent to c_0:

$(c_1 \wedge c_7 \wedge \neg c_8) \vee (c_1 \wedge \neg c_7 \wedge c_8) \vee (\neg c_1 \wedge c_7 \wedge c_8)$

One disjunction of three conjunctions (clauses)

Alternative:

at-least(2, $\{c_1, c_7, c_8\}$, true)

at-least(1, $\{c_1, c_7, c_8\}$, false)

Representing Knowledge: Not-So-Easy Clauses

2	c_1					
c_7	c_8					
	c_{29}	c_{30}	c_{31}			
	c_{36}	**4**	c_{38}			
	c_{43}	c_{44}	c_{45}			

Exactly 2 bullets are adjacent to c_0:
at-least(2, $\{c_1, c_7, c_8\}$, true)
at-least(1, $\{c_1, c_7, c_8\}$, false)

Exactly 4 bullets are adjacent to c_{37}:

$(c_{29} \wedge c_{30} \wedge c_{31} \wedge c_{36} \wedge \neg c_{38} \wedge \neg c_{43} \wedge \neg c_{44} \wedge \neg c_{45}) \vee$
$(c_{29} \wedge c_{30} \wedge c_{31} \wedge \neg c_{36} \wedge c_{38} \wedge \neg c_{43} \wedge \neg c_{44} \wedge \neg c_{45}) \vee$
…
$\vee (\neg c_{29} \wedge \neg c_{30} \wedge \neg c_{31} \wedge \neg c_{36} \wedge c_{38} \wedge c_{43} \wedge c_{44} \wedge c_{45})$

One disjunction of $\binom{8}{4} = 70$ conjunctions (clauses)

<u>Alternative</u>:
at-least(4, $\{c_{29}, c_{30}, c_{31}, c_{36}, c_{38}, c_{43}, c_{44}, c_{45}\}$, true)
at-least(4, $\{c_{29}, c_{30}, c_{31}, c_{36}, c_{38}, c_{43}, c_{44}, c_{45}\}$, false)

Representing Knowledge: Too Many Clauses

2	c_1					
c_7	c_8					
	c_{29}	c_{30}	c_{31}			
	c_{36}	**4**	c_{38}			
	c_{43}	c_{44}	c_{45}			

16 bullets

Exactly 2 bullets are adjacent to c_0:
at-least(2, $\{c_1, c_7, c_8\}$, true)
at-least(1, $\{c_1, c_7, c_8\}$, false)

Exactly 4 bullets are adjacent to c_{37}:
at-least(4, $\{c_{29}, c_{30}, c_{31}, c_{36}, c_{38}, c_{43}, c_{44}, c_{45}\}$, true)
at-least(4, $\{c_{29}, c_{30}, c_{31}, c_{36}, c_{38}, c_{43}, c_{44}, c_{45}\}$, false)

Exactly 16 bullets are in $\{c_0, c_1, \ldots, c_{48}\}$:
One disjunction of $\binom{49}{16} > 3 \times 10^{12}$ conjunctions (clauses)
<u>Alternative</u>:
at-least(16, $\{c_0, c_1, \ldots, c_{48}\}$, true)
at-least(33, $\{c_0, c_1, \ldots, c_{48}\}$, false)

Reasoning with At-Least Constraints

- AtLeastDLX:
  - Draws inspiration from Donald Knuth's exact cover problem "Dancing Links (DLX)" reasoner.
  - Each literal/variable has a column in a doubly-linked sparse matrix.
  - Each at-least constraint has a row in a doubly-linked sparse matrix.
  - Each time a recursive depth-first search of variable assignments assigns a variable to a value, the corresponding column entries are (temporarily) removed and each row is updated.
  - Each time an at-least constraint is satisfied, the corresponding row is (temporarily) removed.

Human Reasoning Steps

General and Definition

Our approach is to create a solver that solves these puzzles with a human-like approach, study the puzzles with their human-like solutions, learn what we can about the characteristics of solutions we enjoy, and then optimize designs for those characteristics.

For this work, we categorize reasoning steps into three types:

- Type 0 Deductions involving a single clue.
- Type 1 Deductions involving multiple clues without knowledge of the total number of bullets.
- Type 2 Deductions involving multiple clues with knowledge of the total number of bullet.

Our solver always seeks a deductive step of the lowest number type, preferring simplest-type next steps in the solution.

Type 0: Single Cell Clue

Type 0 reasoning is the simplest step.

- For type 0 reasoning, we simply iterate through our clues that have adjacent unknown cells, looking for one for which all unknowns must have all or no bullets in order to satisfy the clue’s constraints. Finding one, we mark all adjacent unknowns as known and store information of this type 0 step.

Type 1: Multiple Cell Clues

This is the most computationally complex step of our solver.

- Construct a solver that attempts to find a deduction using only this subset of cell variables and the bullet/no-bullet knowledge of those cells adjacent to them.
- Choose the first possible deduction in grid reading-order, testing the clues sequentially for whether they can be removed from the clue list.
- Test whether further deductions are possible in other unknown cells from the minimal clue subset.
- All type 1 steps have a minimal clue subset and a resulting deduction set, the information of which are stored in our algorithms.

Type 2: Multiple Cell Clues and Number of Bullets

Type 2 deduction is our final resort.

- When a type 2 step is called, all remaining bullet positions will be deduced. Given that we only apply our solver to puzzles with known solutions, this final solution step asserts the only possible configuration of remaining bullets, marks all cells as known, and stores information of this type 2 step.

Puzzle Optimization

Stochastic Local Search

- We optimize a puzzle by treating bullet placement as a combinatorial optimization problem and applying simple stochastic local search in the form of hill descent on an energy (i.e. objective) function that is a numeric measure of puzzle badness according to our subjective puzzle tastes.

- The step function moves one bullet to a different non-bullet position by swapping a random pair indices from the first b entries and the last $n^2 - b$ entries in the aforementioned bullet-cell/non-bullet-cell list.

- The new state results in new clues, new sampling of minimal clue subsets, and new energy computations.

Energy Function

The energy function is given as follow:

$$(0 \times n0) + (-1.5 \times n1) + (n2 \times (n2 - 3))$$

In which,

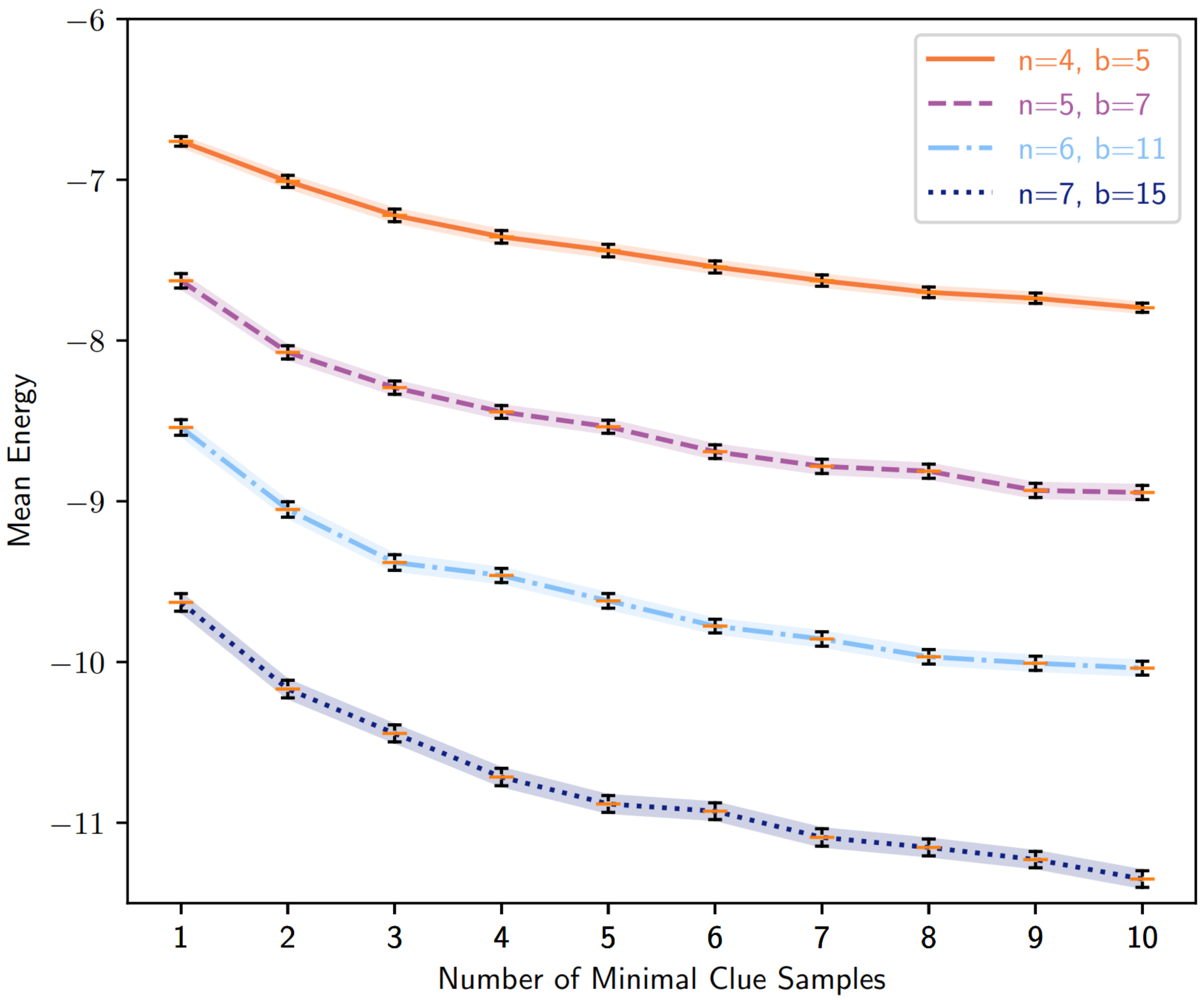
- n0 is the number of deductions requiring the knowledge of a single clue and its adjacent cells.
- n1 is the number of type 1 deductions requiring the interaction of multiple clues.
- n2 is the number of bullets deduced at the end using the constraint of the known number of bullets in the grid.

Sampling Minimal Clue Sets

Sampling Method

- To evaluate our puzzle generation, we opt for a sampling approach. As our sample size increases, we better approximate the optimal energy for the given bullet configuration.

- In this section, we vary the sample size to measure how this affects puzzle quality.



Sampling Results

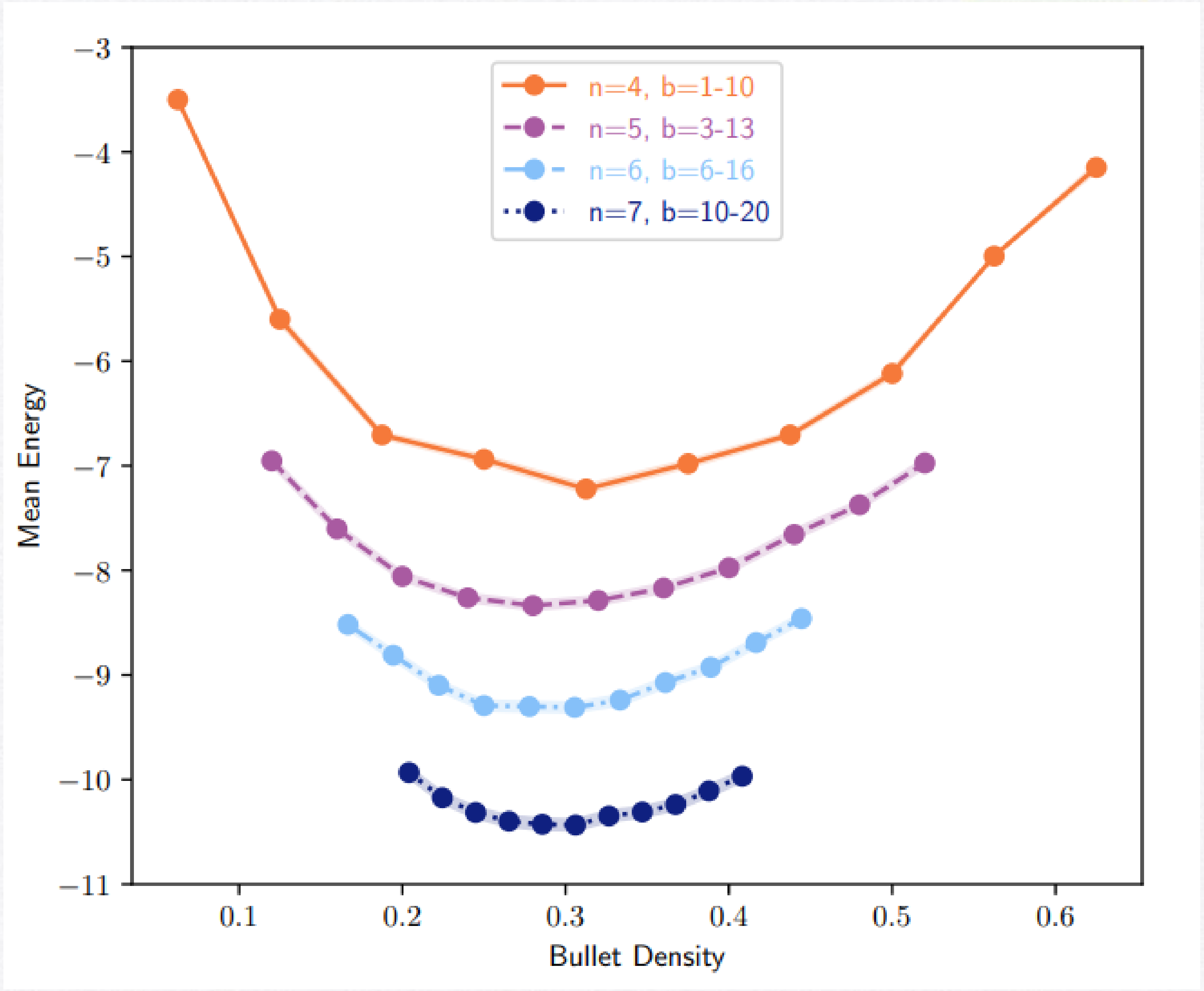
- As demonstrated by the graph, the puzzle quality increases as expected (with decreased energy) when we sample a greater number of minimal clue subsets. Of practical interest is the fact that, for $4 \leq n \leq 6$, much of the quality gain is from the first 3 samples.

- One's application constraints in computational time and space should inform the best trade-off. Generating a few, small high-quality puzzles would suggest a higher number. Generating larger puzzles with limited computational resources would suggest a smaller number.

Bullet Density

Bullet Density Experiment:

- Next, we aim to see how our puzzle compares to computational complexity from a practical puzzle generation perspective.
- A bullet density of approximately 0.3 appears ideal for all grid sizes for our energy function.
- The larger the puzzle grid size, the less sensitive the puzzle quality is to the number of bullets.



Future Work

There are two primary ways we would like to extend this work:

1. Generate high quality, no-guess Minesweeper puzzles.

2. Explore what’s enjoyable about the Bullet Puzzle by further refining our Bullets Puzzle’s classification of human reasoning steps.