

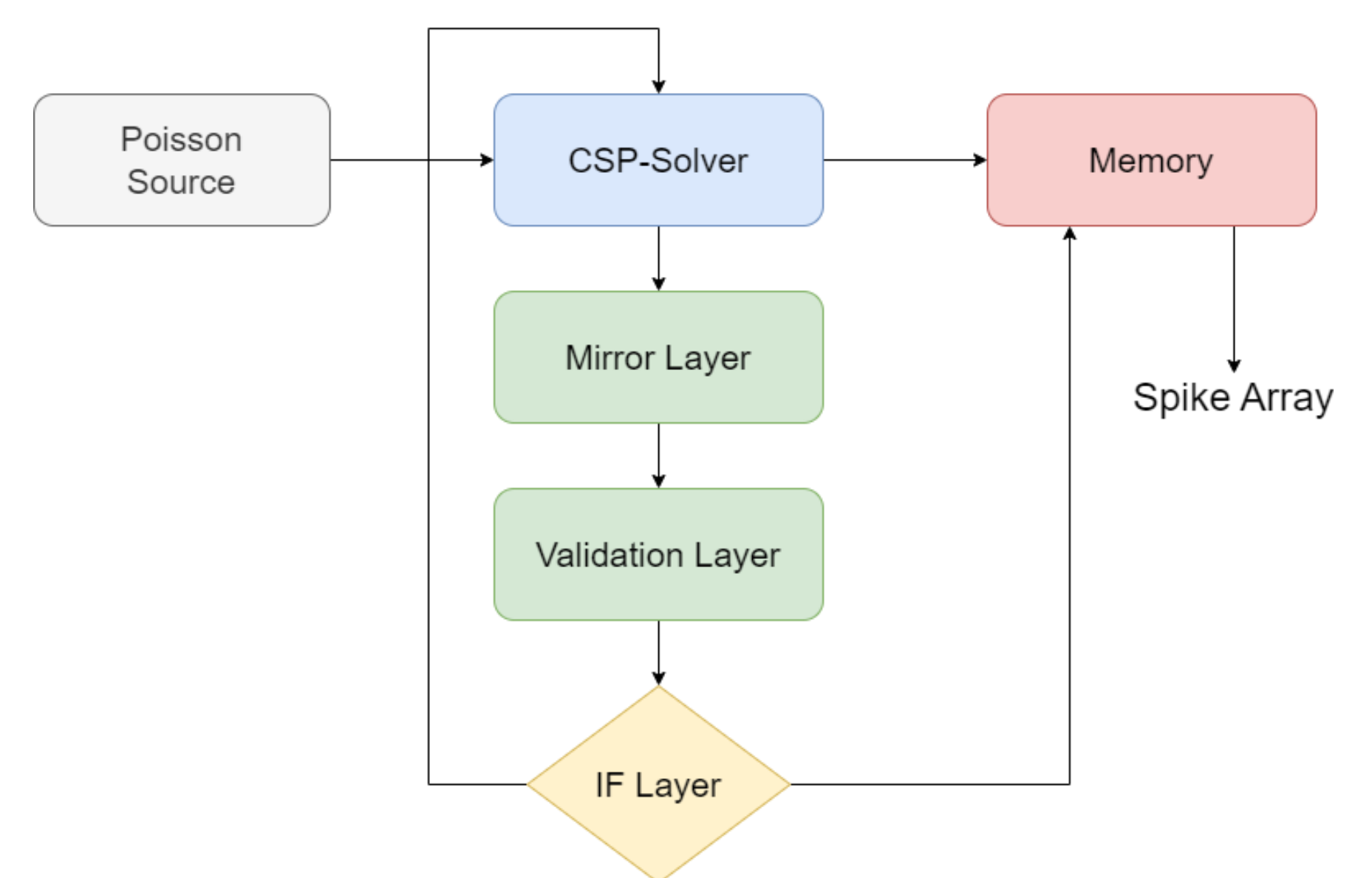
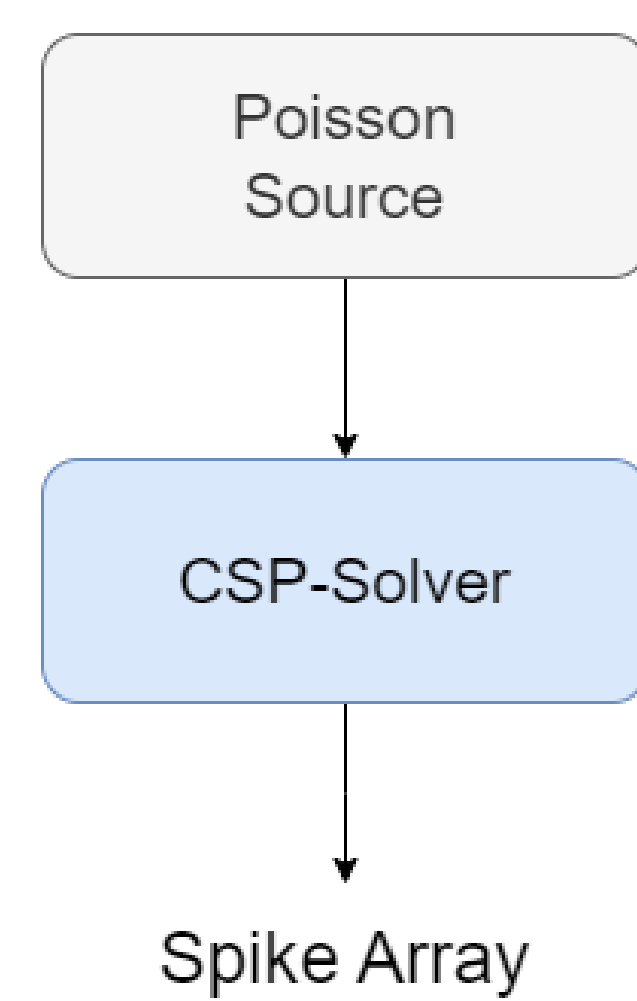
Constraint Satisfaction Problems solution through Spiking Neural Networks with improved reliability: the case of Sudoku puzzles

Description

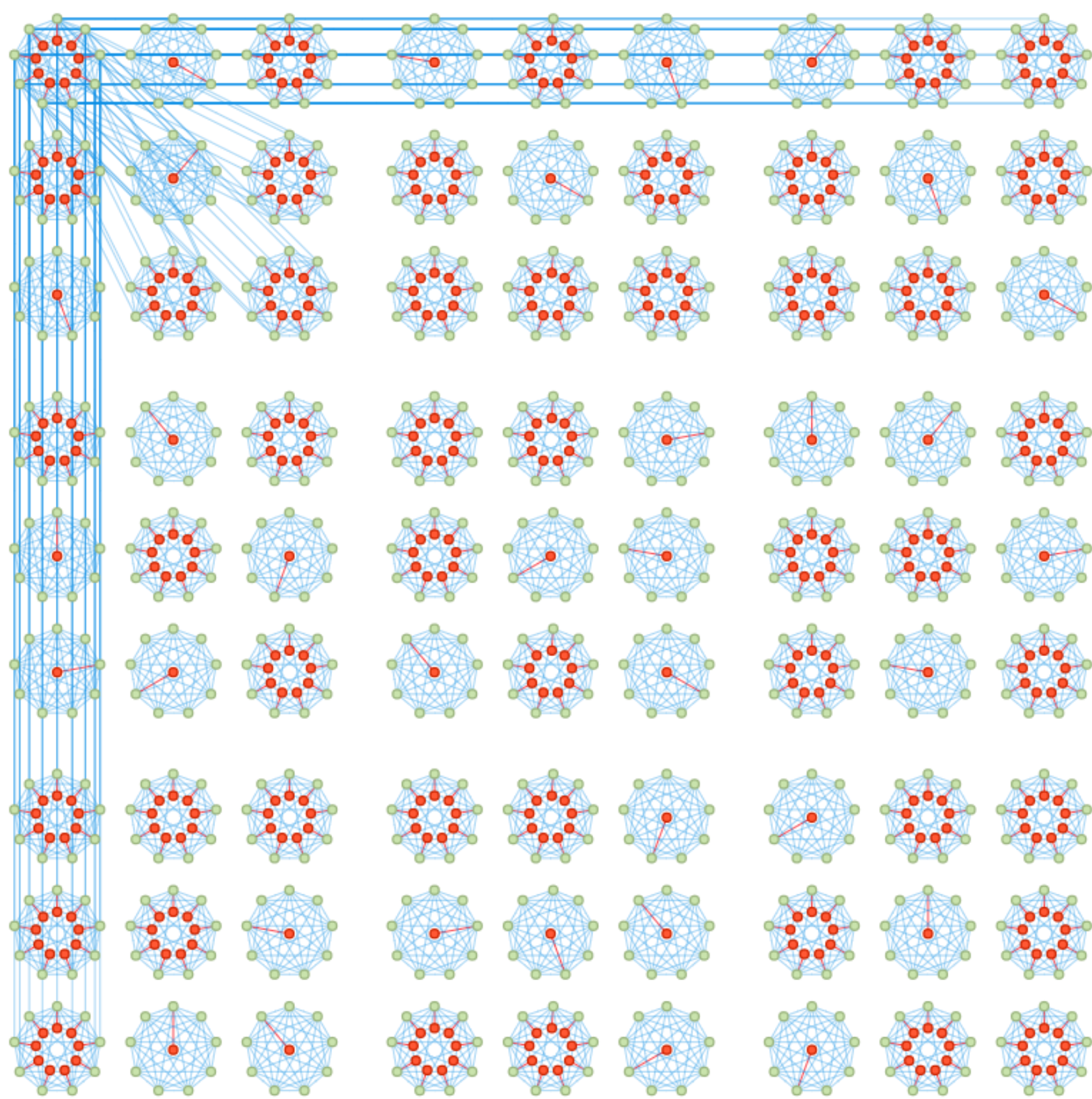
This work discusses about the **Constraint Satisfaction Problems (CSPs)**, specifically the Latin Square Problem (similar to Sudoku puzzles). A proposed solution involves using a stochastic Spiking Neural Network (SNN) made of Leaky Integrate-and-Fire (LIF) neurons [1]. Obtaining a network that stochastically evolves towards the solution, following the **attractor dynamics**, in the configuration space [2,3].

However, the current approach has limitations, such as requiring a pre-set simulation time for each solution attempt and relying on an external platform for validation. Our work propose a new **fully spiking pipeline** that can find, validate, and stop the generation of spikes in the solution process,

	4		8		5	2		
	2			4			5	
5								4
	9				3	1	2	
1		6		7	8			3
3	7		9		4		8	
					6	7		
		8	3	5	9		1	
1	9			7	6			



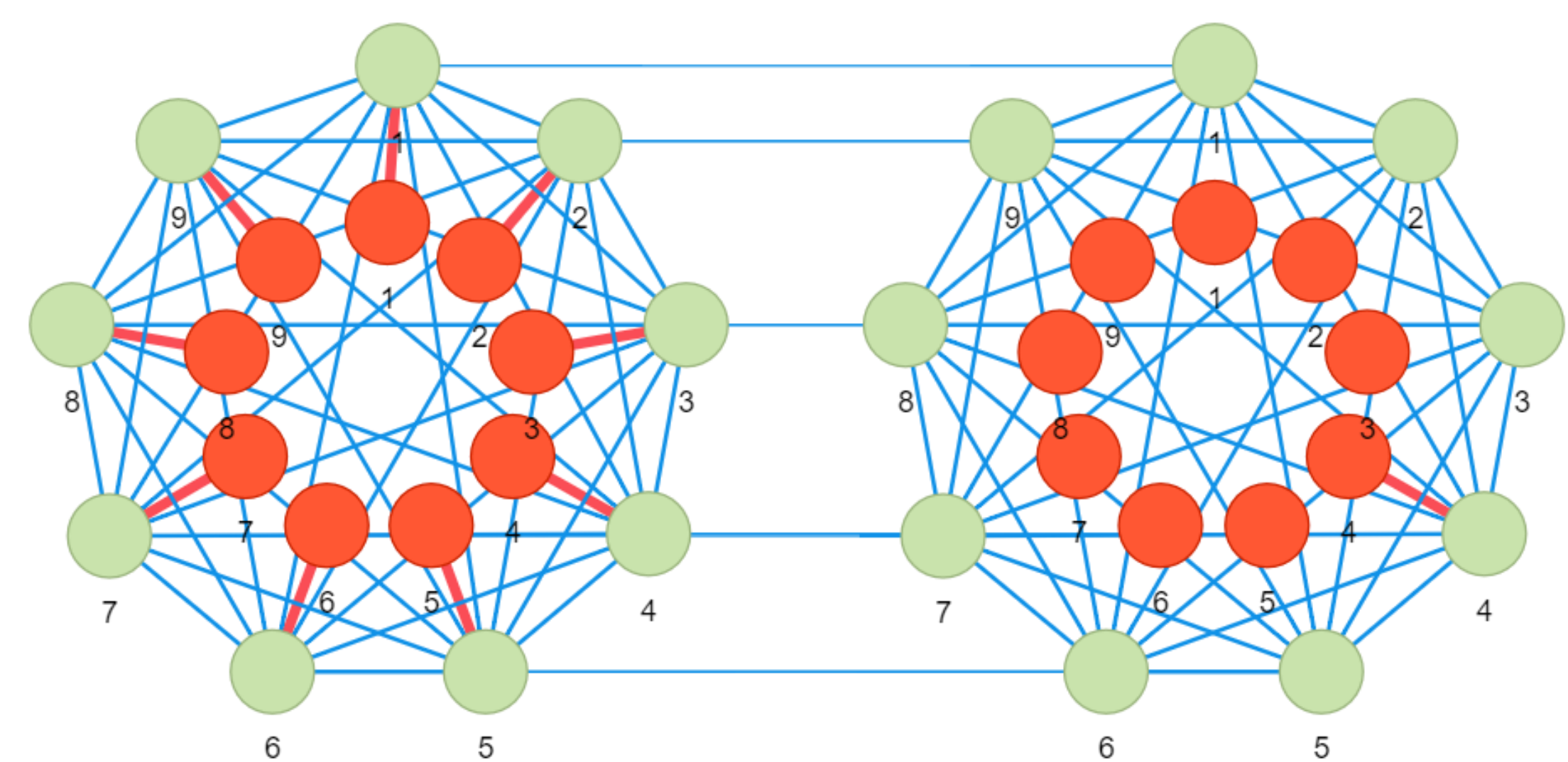
Implementation



The SNN structure is defined by the CSP's mathematical formulation. Each puzzle cell has **9 neuron populations** representing **each possible values**. **Constraints** are modeled through **lateral-inhibition**, connecting populations across cells. This structure forms a sparsely connected graph, encapsulating the problem's formulation with neurons and connections.

•	4	
	2	
5		

The connection structure of adjacent cells, one empty and one with a clue, is shown below. Each cell has **9 neuron populations (green)** in a winner-take-all setup (**blue**). Possible cell values are linked to stimulus populations via excitatory synapses (**red**).



Conclusion

In fig. a), our pipeline outperforms the original approach, achieving a **gain of over 9%** in solutions found and validated. This improvement is attributed to Clue-Hold, maintaining the original CSP formulation during simulation. In fig. b), spiking activity is shown. Our implementation **reduces total spikes by about 3x**, allowing deactivation of the main network after finding a solution. Starting from [1], we have demonstrated an efficient, reliable, and fully spiking solution for Sudoku-like CSPs.

