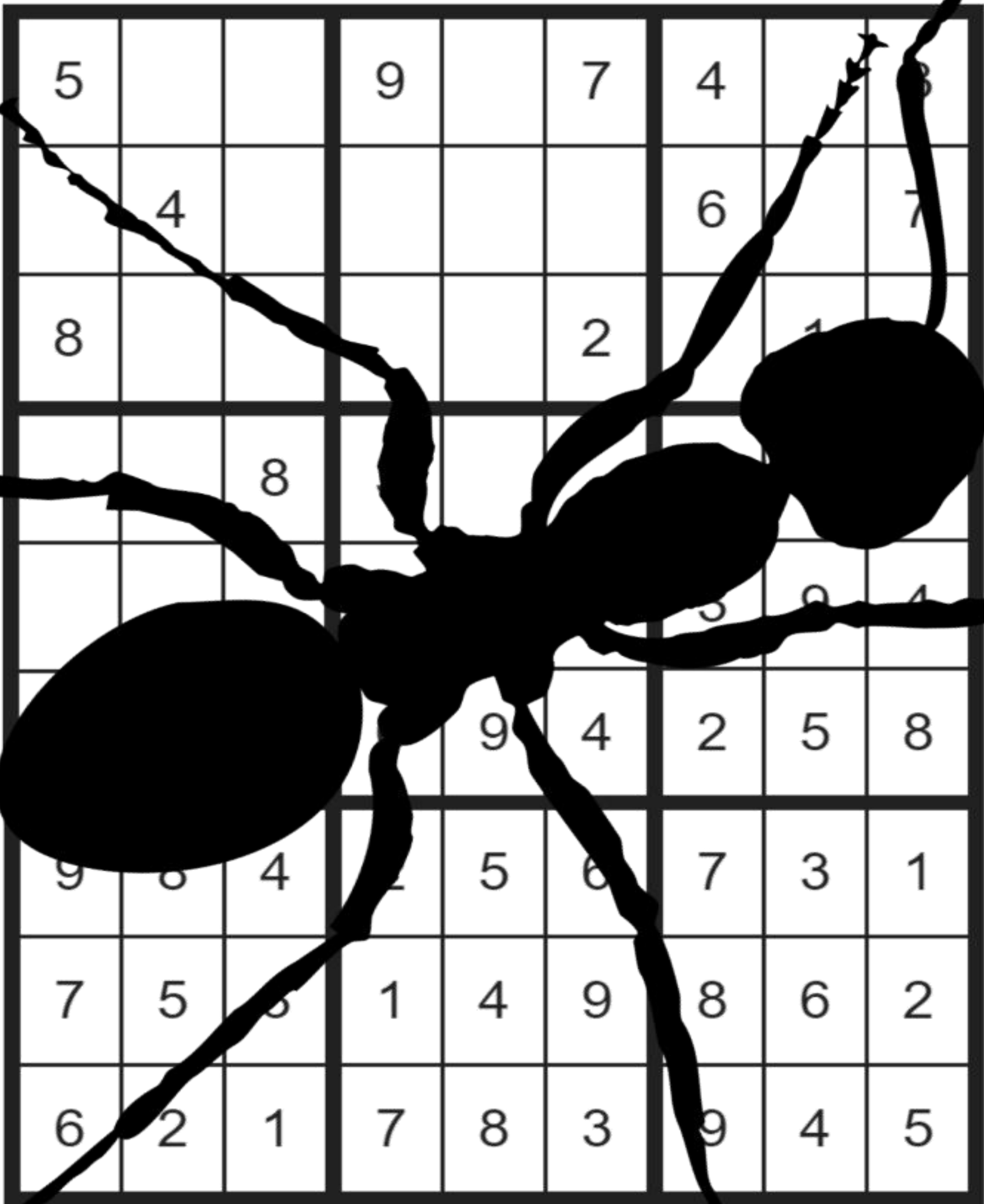


SOLVING SUDOKUS WITH ANT COLONY OPTIMISATION

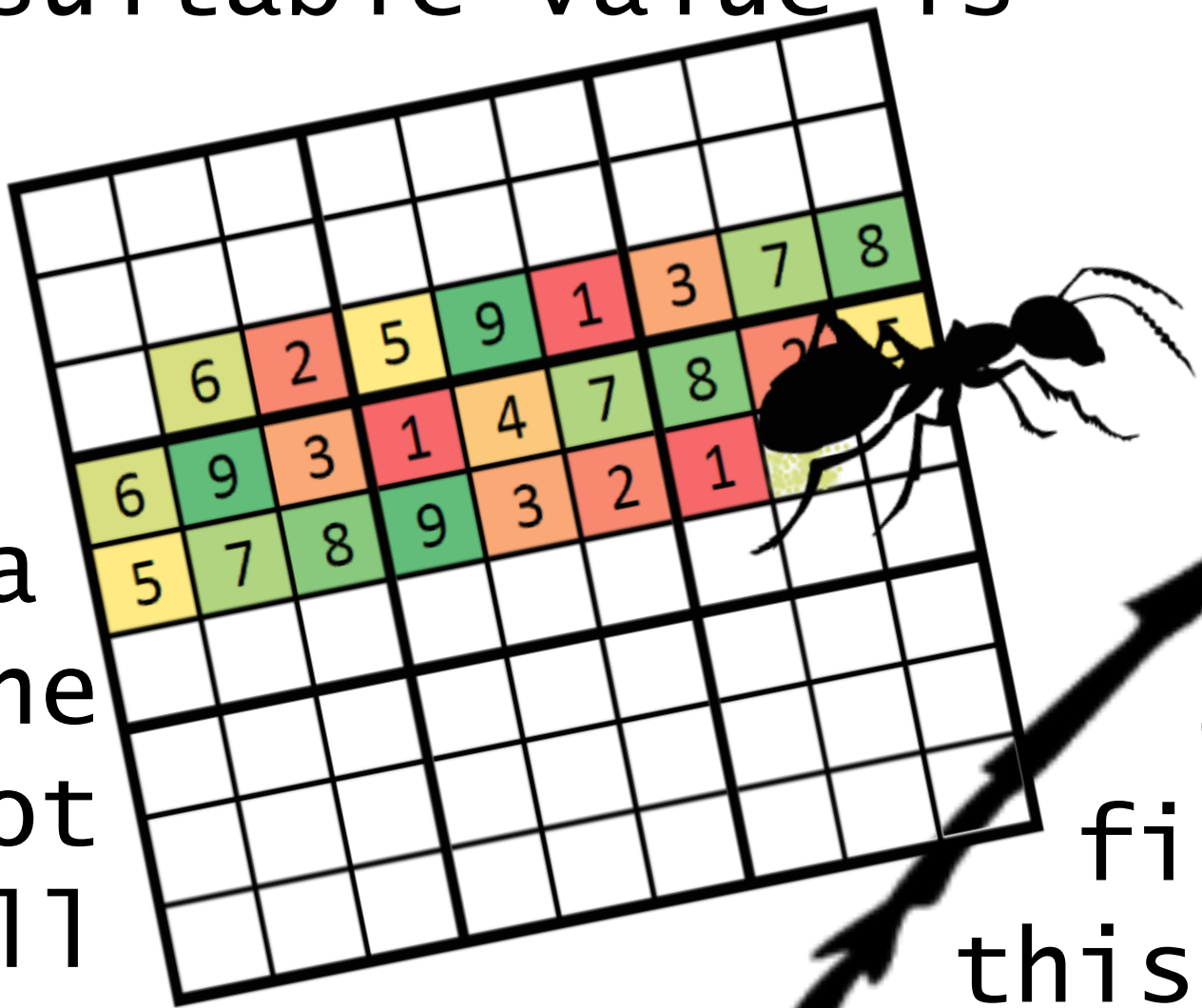
Jüri Gramann, Raid Vellerind
University of Tartu
Algorithmics Course
Fall 2018/19

For solving Sudokus with ant colony optimisation, we give each ant a copy of the Sudoku and let them randomly guess witch value should be in which cell. Most ants will be drawn towards the best solution by global pheromone trail.



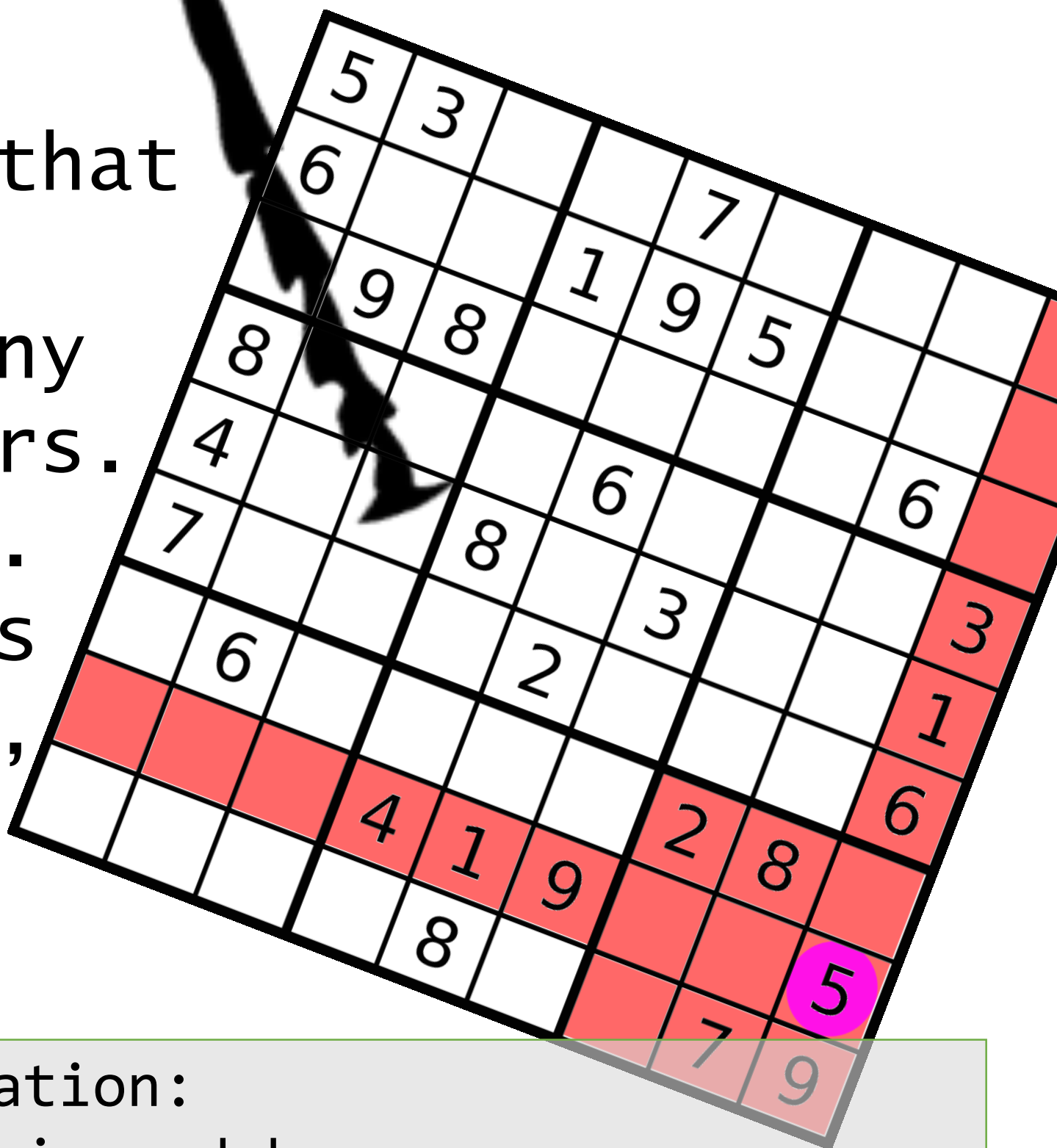
5	0	0	9	0	7	4	0	3
0	4	0	0	0	0	6	0	7
8	0	0	0	0	2	0	1	0
0	0	8	3	0	0	0	7	0
0	0	0	0	7	0	0	0	0
0	3	0	0	0	4	2	0	0
0	8	0	2	0	0	0	0	1
7	0	3	0	0	0	0	6	0
6	0	1	7	0	3	0	0	5

Global pheromone map is initialised with default probability for each value in given cell's value set. With every generation the global pheromone value is updated based on the findings of "best ant." The cell value will most probably be the one with the highest pheromone level. If for any cell the pheromone level has not changed from default, the choice for a suitable value is made randomly.



Each ant starts its journey from a random cell. If the cell's value is not fixed, the ant will make a weighed decision based on the pheromone levels of given cell. After each operation, constraint propagation is applied to the Sudoku to determine which effect the ant's decision had on the whole Sudoku. For an empty Sudoku, an ant would start randomly assigning values and excrete pheromone.

Constraint propagation is a method that fills in all the "obvious" cells. For this the peers of a cell are checked. Any cell has exactly three units and 20 peers. The three units are row, column and box. The set of peers is made up of the cells containing in units. For easier Sudokus, constraint propagation is enough to solve the whole Sudoku.



The final solution of the Sudoku emerges when the described bits and pieces come together. To solve a Sudoku, the first step is to propagate constraints. Then the global pheromone map is initialized and ants are generated. Each ant receives a copy of the Sudoku and starts to travel through it. Based on "best ant's" result, the global pheromone map is updated. New ants from next generation use this as their guide, until the Sudoku is solved

```
1.solve_sudoku():
2.   initialise_sudoku
3.   constraint_propagation
4.   initialise_global_pheromones
5.   while sudoku not solved:
6.       generate ants
7.       each ant tries to solve sudoku
8.       find best ant
9.       update global pheromones
10.      if best ant has solution
11.          break
12.      return solved sudoku
```

```
1.constraint_propagation:
2.   while changes in sudoku:
3.       Eliminate from a cell's value
4.       set all values that are fixed
5.       in any of the cell's peers.
6.       If any values in a cell's value
7.       set are in the only possible
8.       place in any of the cell's units,
9.       then fix that value.
10.      return changed sudoku
```