# *FIT3139: Lab questions for week 11*

## *Question 1*

In this exercise, you will write a script which will attempt to solve the $N$ Queens problem using (1) simulated annealing, and (2) a genetic algorithm.

*Description of the N Queens problem:* You are given a $N \times N$ chessboard and $N$ queens. In chess, a queen from a given position on the board can move as far as she wants, horizontally, vertically or diagonally. The problem is to find a placement of $N$ queens on the $N \times N$ chessboard such that no two queens bump into each other in a move.

Write a script for this problem using both the heuristics (simulated annealing and genetic algorithm) taught to you during week 10. Your script should accept $N$ as a parameter. Your energy/cost function for a given configuration of $N$ queens on the board is the number of pairs of queens that are in conflict. Your program should terminate when the number of conflicts becomes zero and report the configuration of the queens.

## *Question 2*

Write a script that attempts to solve an order-$n$ Sudoku puzzle, using (1) Simulated annealing and (2) Genetic Algorithms.

*Description of Sudoku Puzzle:*

The object of sudoku is simple: given an $n^2 \times n^2$ grid divided into $n \times n$ distinct squares (see the squares with bold lines in Fig. 1), the aim is to fill each cell so that the following three criteria are met:

1. Each row of cells in the $n^2 \times n^2$ grid contains the integers 1 through to $n^2$ exactly once.
2. Each column of cells in the $n^2 \times n^2$ grid contains the integers 1 through to $n^2$ exactly once
3. Each $n \times n$ square contains the integers 1 through to $n^2$ exactly once.

(See Fig. 2 for a solution to the puzzle in Fig. 1)

Your script must accept an $n^2 \times n^2$ puzzle in a text format. This is simply a matrix with certain cells filled and the rest empty (represented by some dummy symbol). Your script should output a solution to the puzzle if it has been able to find one.

Among the things you should consider when developing this script are:

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

Figure 1: An order-3 sudoku puzzle that can be easily solved

- a scheme for an initial configuration (before the start of the simulated annealing) where the third of the three criteria listed above is automatically satisfied.
- a cost function which has to be minimized to achieve a solution to a given puzzle.
- a scheme for perturbation of any given configuration

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *1* | **2** | **4** | *9* | *5* | **7** | *3* | *8* | *6* |
| **6** | *8* | *5* | *3* | *4* | *1* | *2* | *9* | *7* |
| *9* | *7* | **3** | **6** | **8** | *2* | **4** | **1** | **5** |
| **4** | **3** | **1** | *2* | *6* | **5** | *9* | *7* | *8* |
| **5** | *6* | *8* | *4* | *7* | *9* | *1* | **3** | **2** |
| **7** | **9** | *2* | *1* | *3* | *8* | *5* | **6** | *4* |
| **2** | *5* | **9** | **7** | **1** | *6* | **8** | *4* | *3* |
| *8* | **4** | *7* | *5* | **9** | **3** | *6* | *2* | *1* |
| **3** | **1** | *6* | *8* | *2* | **4** | **7** | **5** | *9* |

Figure 2: Solution to the sudoku puzzle in Fig. 1.