ID2209 Distributed Artificial Intelligence and Intelligent Agents

Assignment3 – Coordination & Utility

Group 15

Zainab Alsaadi

2021-12-01

In this assignment, The N-Queen problem was solved using backpropagation in the first task. The problem description is that given NxN sized chess board, find all the ways to place N queens, such that no queen is under attack.
The second task was about calculating utility for guest agents to find the most appropriate stage according to their preferences.
The aim of the tasks is to understand how agents communicate and cooperate to achieve their goal.

How to run

Run Gama 1.8 and import Task1.gaml and Task2.gaml as new project. Press main to run the simulation. You can see the simulation in both the console and the graphical display.

Species

Task1:
Queens: each queen communicates only with predecessor and successor (using fipa). All N queens should position themselves so that no one is under attack. The simulation stops when all queens are correctly positioned.

ChessBoardGrid: the grid cells are used as a base for the queens' positions and has the size NxN.

Task2:
Stage: stages have different attributes, and each attribute has different evaluation. Stage variable values are changed each interval while the guest's stays the same.

Guests: calculates their utility for each stage. The stage with the highest utility is picked and travelled to. They communicate with stages via fipa to know their values.

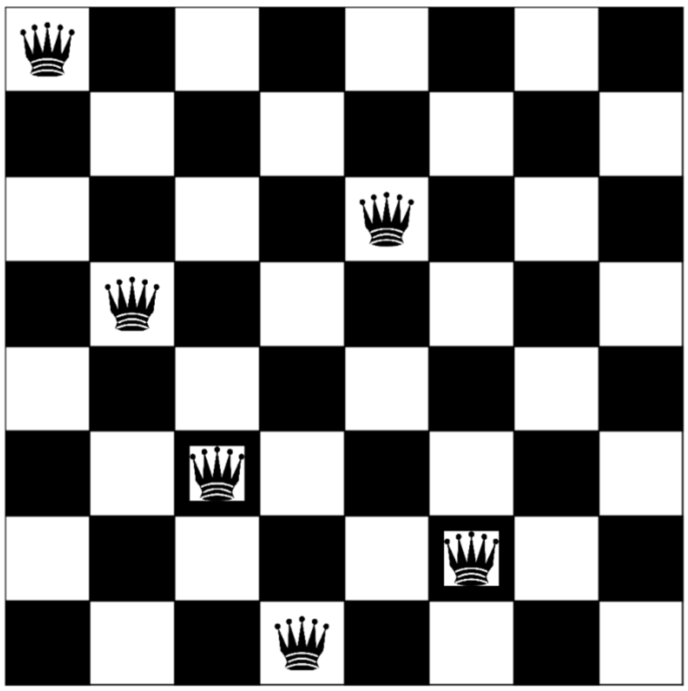
Implementation

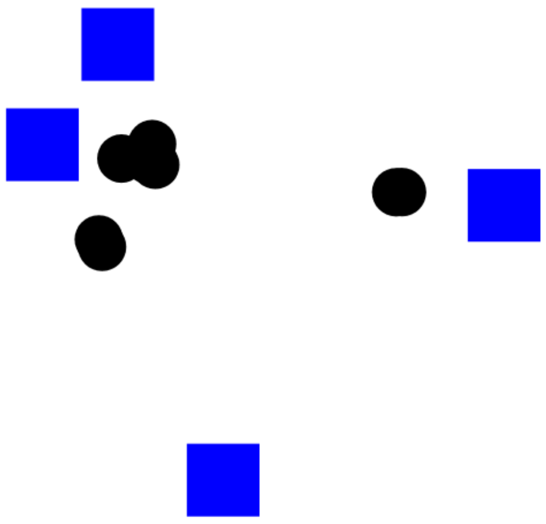
The protocol used in these tasks is *fipa_request* with the *inform* performative. Task1 is implemented by having N queens and a chessboard of a suitable size. The queens start placing themselves at their corresponding column and when a queen's position conflicts with another the previous queen is informed so that it changes its position. When a queen finds a suitable position, it informs the successor queen that it is her turn to find a place. If no place is available, backpropagation guides the predecessor queens into another sequence of positions until the correct one is found.

The second task was implemented by randomly assigning values to stage attributes and guest preferences. The utility function is then calculated for each guest and the highest utility leads the guests to their target stage. The stage regenerates new values after each interval and the guests recalculate their utilities to move to another target, if higher utility is found.

Results

Below are screenshots from the two tasks.


Task1, during repositioning of queens.


Task2, guests standing at their most preferred stage.

Discussion/ Conclusion

Task 1 was interesting, challenging and fun. Task 2 was less interesting. The extra challenges have not been looked at because of the lack of time, in addition to that the assignment was too time consuming and too much work had to be done.