# Assignment 2

# STE6246-002

Machine learning for multi Agent System

## Abstract

The assignment 2 was about the multi agent system for machine learning approach. I have referred this for the codes and works of Bjørn, Fati and Ghada, project.

This is an auction trade between two smart house agents A and B controlling the heating of the rooms in different temperature environment. For A this interval is 23 to 25 degrees and for B this is 20 to 23. If they can be within the given range they will get a reward if they can't then they will get penalty. And here comes the third agent C who is responsible for the energy ceiling value determination for A and B. The agent will also monitor the solar panel which is simple to more sun more energy. If more energy is generated then the ceiling value for the A and B can be raised or vice versa if energy is low.

A market has a setup for this process, where Agent C will hold an auction to raise or lower the ceiling value for a called price. A and B have to make demand for their energy. If the total volume exceeds then the demand must be lowered down with a call. If the ceiling is lowered, then the demand must get up.

When the sum of the bids is all Pareto efficient, i.e. the sum of the volume is at the ceiling or slightly below it, the process stops. The assignment is split into two questions and three parts.

## Questions

1. Which of the two agents will be better off (more credits) than the other, if any?
2. Will agent C be able to maintain its ceiling at all times?

Parts:

A. Design a system model (architecture) and a brief specification on how you would like this system to behave. Simple UML representations would be preferred. (Paper exercise)

B. Implement the model and carry out different simulations to answer the main questions above? (Program)

C. Show how Q-learning could be used for anyone of the two agents A, B to outperform the other. Illustrate the R matrix and example Q-matrix for this. (Paper exercise)

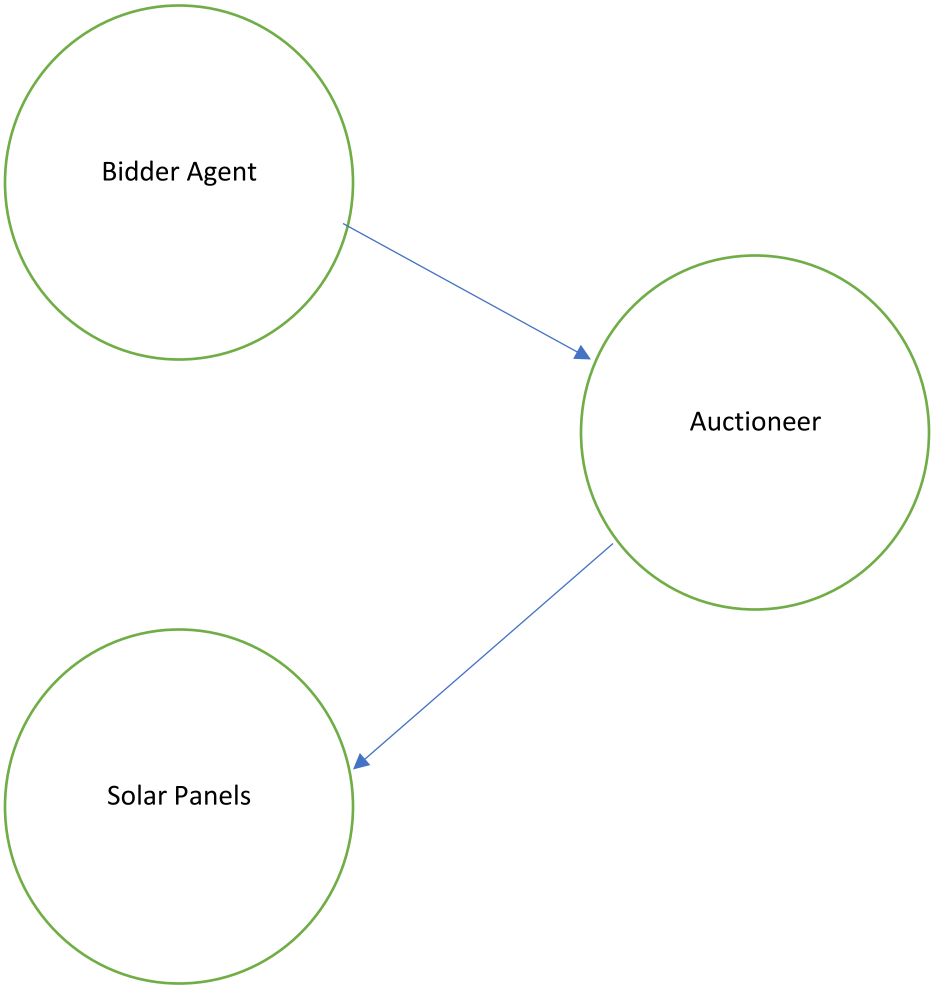


*Figure 1: Simple Model*

## Programming

As in this project assignment, it is to create the model and to sun the simulation to propose the answer of the questions asked. How does this work; creating 3 agents A, B and C. and a solar panel model class to provide energy. The main idea is to run the program until the one of the agent's credit is gone completely.

A and B has the defined temperature range which if they maintain they will get reward otherwise they will be penalized, and the update of ceiling is according to the agent c, which dues with the change in weather condition affecting solar panels.

### Methodology

The main methods to implement this is to follow the Bellman's equation

$U(s, a) = R(s, a) + \alpha * M ax[P(s 0 , a0 ) * U(s 0 , a0 )]$

This is a utility tool to calculate the state which is equal to reward and discount factor alpha is multiplied to the maximum utility of the next state times of the probability. This continues for the next states. Now in this assignment using a discount factor of 0.7 and probability of moving in specific direction was 0.5, 0.2 and was assigned for stay, up and down.

## Energy generation

The energy generation is created as per description, the value of the ceiling must be increased as if the energy needs of the agents were larger than anticipated. The ceiling will be between 11 and 19KWh and the energy requirements for the agents going from their absolute minimum temperature of 21 and 18 to 22 and 19, still outside of their" safe" interval was between 50 and 60KWh for a time of 6 hours for both agents. Therefore, the value of the ceiling was multiplied by 6 to compensate this approach.

## Results

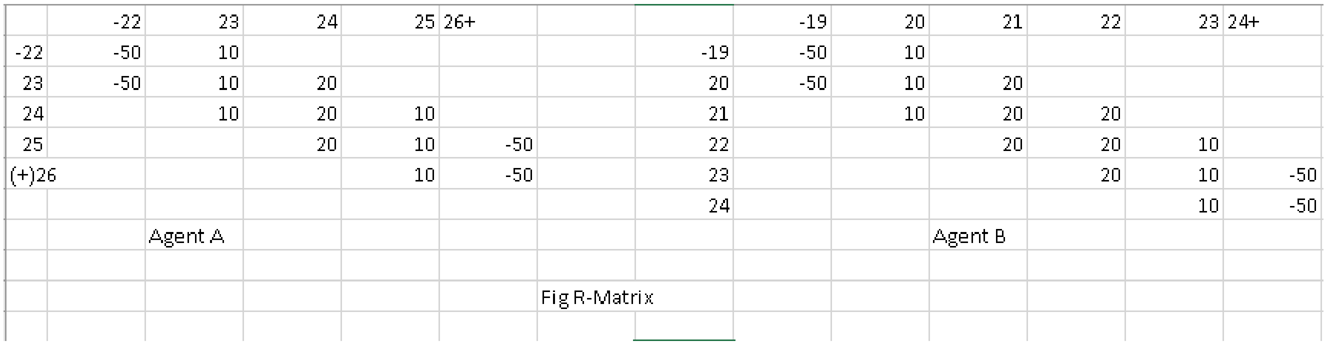
The simulation has altering rewards of the agents and the ceiling, it seems like agent B ends up with more credits than agent A most of the time. This has some logical meanings as agent B has a wider interval of temperatures that gives rewards rather than penalties and needs to maintain a lower temperature. On the other hand, agent, A has a cheaper energy consumption rate, but the difference turns out to be negligible. It is also because agent C will not be always being able to maintain its ceiling. The agents receive a higher reward for staying towards the higher end of their temperature interval and with the combination of low outdoor temperatures and a solar panel that is only active for 12 hours agent's A and B often asks for more energy than agent C can provide, and it will therefore try to initiate a trade between the two bidders or increase the price of the energy it sells.

## Q-learning

Q-learning could be used for anyone of the two agents A or B to out perform the other.  But our work on this assignment and conversations with Prof. Bremdal has taught us that these two selfish agents will learn over time that the best result for each of them will be achieved through cooperation with the other agent. To illustrate this with Q-learning, an attitude of cooperation was the starting point of this part of the task. For this task a static R matrix has been assumed, meaning no alterations caused by trading or anything else. Legal movement between states has been defined as back one, stay, or forward one. If multiple states have to be traversed, for example Q(24, 21) then that will be split into multiple ”one step” equations, i.e Q(22, 21)...Q(23, 22)...Q(24, 23).

## R Matrix

Each agent starts with different R matrix, but here they have been merged together to one matrix with the values determined by an expression utilizing values from the previous, individual matrices RA+RB/ 10 . The agents will each have their own personal Q matrices, so any interaction will come through this shared R matrix instead.



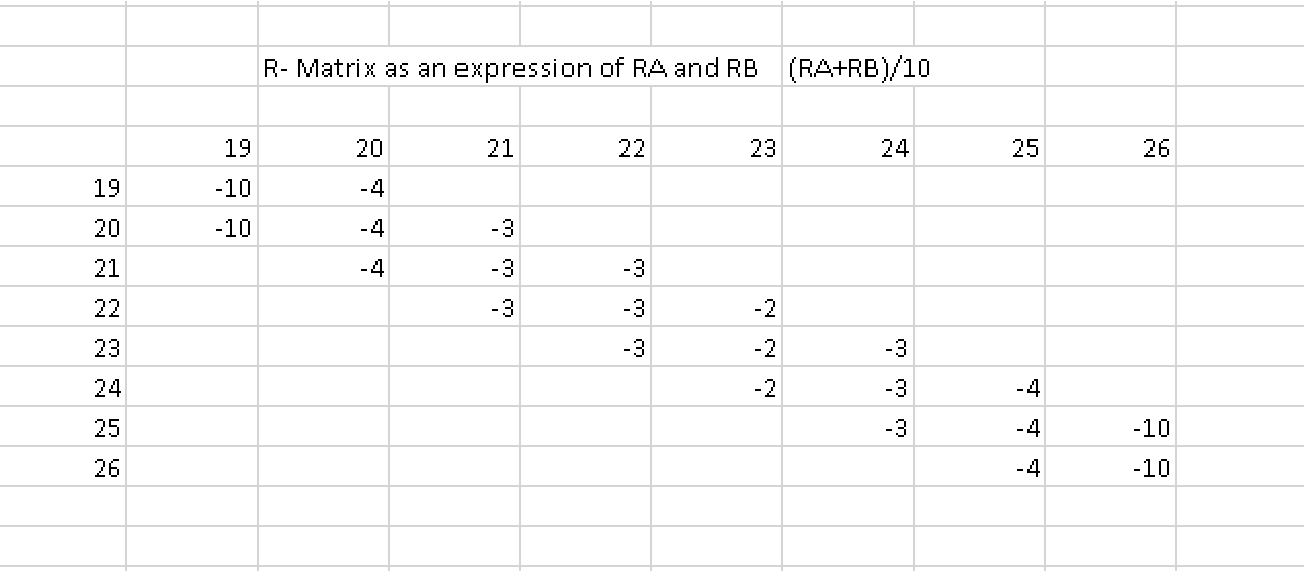
|  | -22 | 23 | 24 | 25 | 26+ |
|---|---|---|---|---|---|
| -22 | -50 | 10 |  |  |  |
| 23 | -50 | 10 | 20 |  |  |
| 24 |  | 10 | 20 | 10 |  |
| 25 |  |  | 20 | 10 | -50 |
| (+)26 |  |  |  | 10 | -50 |

Agent A

|  | -19 | 20 | 21 | 22 | 23 | 24+ |
|---|---|---|---|---|---|---|
| -19 | -50 | 10 |  |  |  |  |
| 20 | -50 | 10 | 20 |  |  |  |
| 21 |  | 10 | 20 | 20 |  |  |
| 22 |  |  | 20 | 20 | 10 |  |
| 23 |  |  |  | 20 | 10 | -50 |
| 24 |  |  |  |  | 10 | -50 |

Agent B

Fig R-Matrix



R- Matrix as an expression of RA and RB (RA+RB)/10

|  | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
|---|---|---|---|---|---|---|---|---|
| 19 | -10 | -4 |  |  |  |  |  |  |
| 20 | -10 | -4 | -3 |  |  |  |  |  |
| 21 |  | -4 | -3 | -3 |  |  |  |  |
| 22 |  |  | -3 | -3 | -2 |  |  |  |
| 23 |  |  |  | -3 | -2 | -3 |  |  |
| 24 |  |  |  |  | -2 | -3 | -4 |  |
| 25 |  |  |  |  |  | -3 | -4 | -10 |
| 26 |  |  |  |  |  |  | -4 | -10 |

*Figure 2: Fig R-matrix*

**Iterations:(calculation of Q values)**

QA (25,26)= Q(25,26) +a(R(25,26) +g*max[Q(26.25)) *Q(26,26)]-Q(25,26))

QA(25,26)=  0+0.7(-10+0.3(0,0)-0) = -7

QA(22,23)= 0 +0.7 (2+0.3(0,0,0)-0) = 1.4

QA(26,25)= 0+0.7(-4+0.3(-0.7,0,0)-0)= -2.8

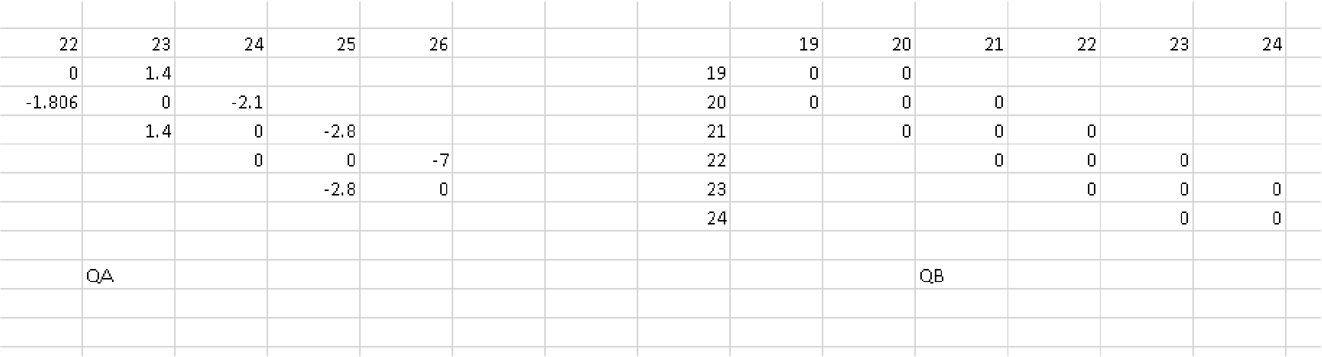
QA(23,22)= 0+0.7(-3+0.3(0,0,1,4)-0)= 0.7(-3+0.42) = -1.806

QA(23,25):

QA(24,25)= 0 +0.7 (-4+0.3(0,0,-0.7)-0) = -2.8

QA(23,24)= 0 +0.7 (-3+0.3(0,0,-2.8)-0) = -2.1

QA(24,23)= 0+0.7(2+0.3(-2.1,0,0)-0)= 1.4



| 22 | 23 | 24 | 25 | 26 |
|---|---|---|---|---|
| 0 | 1.4 | | | |
| -1.806 | 0 | -2.1 | | |
| | 1.4 | 0 | -2.8 | |
| | | 0 | 0 | -7 |
| | | | -2.8 | 0 |

QA

| | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|
| 19 | 0 | 0 | | | | |
| 20 | 0 | 0 | 0 | | | |
| 21 | | 0 | 0 | 0 | | |
| 22 | | | 0 | 0 | 0 | |
| 23 | | | | 0 | 0 | 0 |
| 24 | | | | | 0 | 0 |

QB

*Figure 3: QMatrix*