COMP6203 Intelligent Agents
Lab Instructions

Jan Buermann
j.buermann@soton.ac.uk

2024

Contents

Lab #4 and Beyond

4.1 Introduction

There are not further exercises. You can use the labs to develop your own agent for the coursework. To help with that you can use the provided competition playground to run bigger examples than those in the previous labs.

4.2 A More Competitive Playground

To make development of your coursework agent easier you can use the following provided files.

```
main_competition_playground.py
groupn.py
```

- Download and extract the files now if you have not done that already.
- Have a look at the code.

Similar to other lab exercises there is an entry point scrip to run your python code, which is `main_competition_playground.py` in this case, and, in preparation for the coursework submission, `groupn.py` is for you to add your agent code.

- Within `groupn.py` there is a class called `Companyn` which is where you should **add your code**.

If you look into `main_competition_playground.py` you should see in the function `def build_specification` that your company and two further companies are added to the simulation. The company `class MyArchEnemy` and `class TheScheduler` operate similar to the agents you developed yourself in the labs.

There are a number of options to adjust so that you can test different scenarios.

- `number_of_month` allows you to specify how long the simulation will run. Every month there will be one auction.
- `trades_per_auction` allows you to specify the number of trades that are being auctioned in one of the monthly auctions.
- `def mixed_fleet` is used to create fleets.
  - It generates fleets with a specified number of three types of vessels called *Suezmax*, *Aframax* and *VLCC*.

  - The function expects three arguments: `num_suezmax`, `num_aframax` and `num_vlcc` to specify the number of the respective types of vessels in the fleet, i.e.

    ```
    trades_per_auction(num_suezmax=2, num_aframax=3, num_vlcc=0)
    ```

    will create a fleet with 5 vessels of which 2 will be *Suezmax*, 3 will be *Aframax* and none will be *VLCC*.

- Both `MyArchEnemy` and `TheScheduler` have a constructor argument `profit_factor` which, similar to Exercise 3 in Lab 3, is used to modify the cost to bid calculation of the agent, i.e. the agent will bid `cost * profit_factor`.

- Also take note of the `global_agent_timeout` parameter of the environment creation function `def generate_simulation`. It is set to 60 by default, which means your company has 60 seconds to complete `pre_inform`, `inform`, `receive`. If your company does not finish within that time its operation will be cancelled and your company will not bid or schedule the trades and thus looses its chance to transport cargoes or will have to pay a penalty.

You are encouraged to play with different scenarios. Feel free to change the length of the simulation, the number of trades per auction, the number of vessels in the fleets, the number of competitors, their profit_factor etc..

As mentioned before, remember that your competitors may not just use a profit factor to determine their bids and may take other cost, like reaching the origin port or idling somewhere, or even your fleet and bids into consideration.