Homework 2 - Autonomous Networking

Giovanni Pica[1], Andrea Bernini[2], and Donato Francesco Pio Stanco[3]

[1] Student Identification Number: 1816394
[2] Student Identification Number: 2021867
[3] Student Identification Number: 2027523

1 Introduction

The problem that we are going to discuss in this report is the *General Scenario Problem*, based on creating a *Reinforcement Learning Routing Protocol* for deciding to keep, send a packet to other neighbours drones or go to the depot (physical delivery). In this problem we have a squad of drones (N $\geq$ 2), and NO ferries. All the drones continuously explore an area of interest to detect and monitor the area, and each drone may have a different speed and has no circular trajectories, in the end during a physical delivery , drones can't sense new data.

We need to deliver as many packets to the repository as possible for the main task and as a secondary task to reduce packet latency by minimizing the final score, but also the energy spent to physical movement toward the depot.

2 Methodology

2.1 Reinforcement Learning Approach

In this section we discuss the approaches to solve the General Scenario Problem. For creating this algorithm we used the Q-Learning Algorithm and the action choice mechanism is based on ϵ-Greedy selection, we generate a random number between 0 and 1, if this number is lesser than epsilon we take a random action, otherwise we take the action with the greater Q_value.

The actions in this algorithm are described as follows:

- **0**: Send the data to a neighbor drone, which may arrive at the depot before him.
- **1**: Store the data and wait to arrive at the depot.
- **2**: Move the drone physically to the depot.

To send the packet we used the simple GeoRouting algorithm, this is because it is less expensive in terms of energy, in fact, if there are no neighbors, it returns None, instead the MoveGeoRouting returns -1 and then performs the action of going to the depot .

2.2 The state representation

The first step was to divide the environment into a grid formed by a number of cells depending on the size of the communication range of the drones (in this case the result was a 4x4 grid, Figure 1). In such a way as to represent the states of our agent through cells, each of which has associated three values (corresponding to the three actions). So for each drone we will have a Q_Value attribute corresponding to a matrix having the three actions as columns and the number of cells as rows.

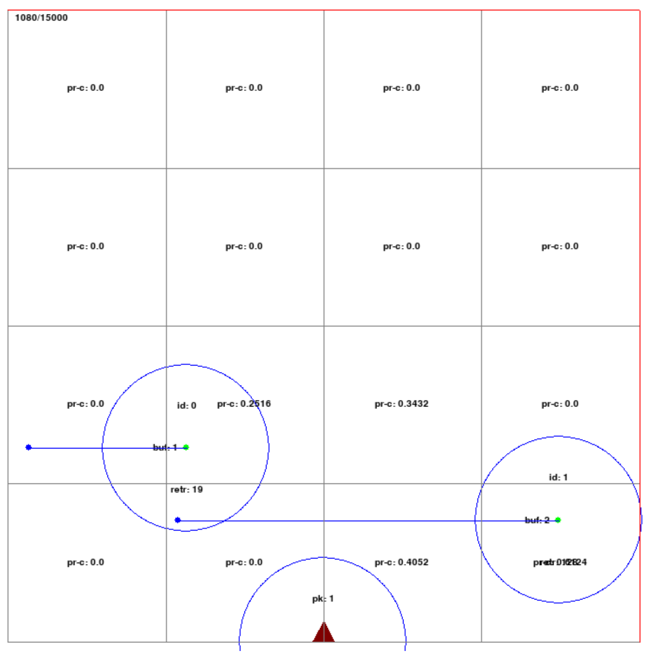

Fig. 1: 4x4 Grid

2.3 Reward

For each action performed, the Q-value is updated according to the formula:

$$Q_{cell,action} = Q_{cell,action} + \alpha(Reward + \gamma * argmax(Q_{cell}) - Q_{cell,action}) \quad (1)$$

The reward is calculated based on the position of the drone, i.e. based on the cell in which it is located and varies from each type of action, according to the following scheme, using constant numbers:

- **Action 2** (move to depot) and **Action 0** (send the packet): The reward decreases as the distance from the depot increases and vice versa. This is because in the case of action 2 a greater distance from the depot is equivalent to a greater waste of energy. While for action 0 it is more convenient to send the packet if I am in a very distant cell, as it is likely that the packet will arrive sooner.
- **Action 1** (keep the packet):The reward increases with increasing distance from the depot and decreases near the depot. This is because if the drone is very close it is likely to enter the communication range of the depot.

Furthermore, when a drone delivers the packages to the depot after performing action 2, the reward used is that relating to the starting cell multiplied by the time necessary for the drone to return to its mission, calculated by dividing the *distance between the depot and the point in which the mission is resumed*, due to the *speed* of the drone.

2.4 Issues

We had some issues including the fact of not being able to overcome the MoveGeoRouting score and so we had to find a better approach and another problem was to handle repeated feedback due to network errors. Other problems were the fact to minimize the energy consumption and the choice of the states.

2.5 Real Scenario

In a real scenario we suppose that the Q-Learning algorithm is better than the Bandit of the first homework because is more suitable and explores not only the stochastic reward function but also the state and state transition probability, so a drone explores many times.

3 Experimental study

In this section we show performances of each approaches:

- RND: Random Routing,
- GEO: Geographical Routing,
- MGEO: Move Geographical Routing,
- **AI**: Q-Learning Routing,
- **AI2**: Q-Learning Routing but applied only on action 0 and 1 and a drone goes physically to the depot when is really near.
- **OLDAI**: Is the approach used in the first homework only to the action 0 and 1 and a drone goes physically to the depot when is really near.
- **OPT**: Optimistic initial values on the main AI routing.

The highlighted algorithms have been developed ourselves.

3.1 Setup

All experiments in this section, unless otherwise specified, 48000 steps (*len_test*), which correspond to three hours of mission. The libraries used by the project are the python3 standard library and Numpy.

3.2 α and γ Choice

As a first step we ran several tests on the AI algorithm, to decide which is the best choice for the α value. We have chosen to print α values quite different from each other to have a better view (0.2, 0.4, 0.6, 0.8). From the graph in Figure 2 it turns out that the best choice is to set α equals to 0.7 because is in the middle of 0.6 and 0.8 also considering the energy consumption. After the α choice we choose γ and we have chosen to print values quite different (0.3, 0.6, 0.9) and from the graph in Figure 4 it turns out that the best value is 0.6 also considering the energy consumption.

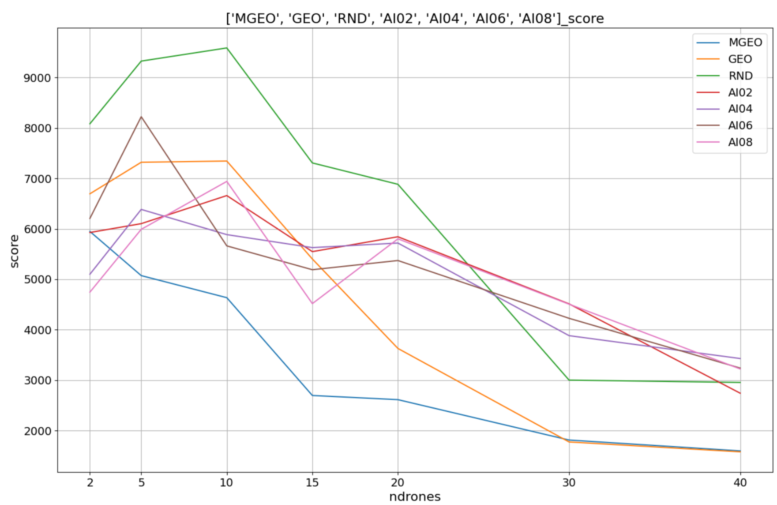

Fig. 2: All α score

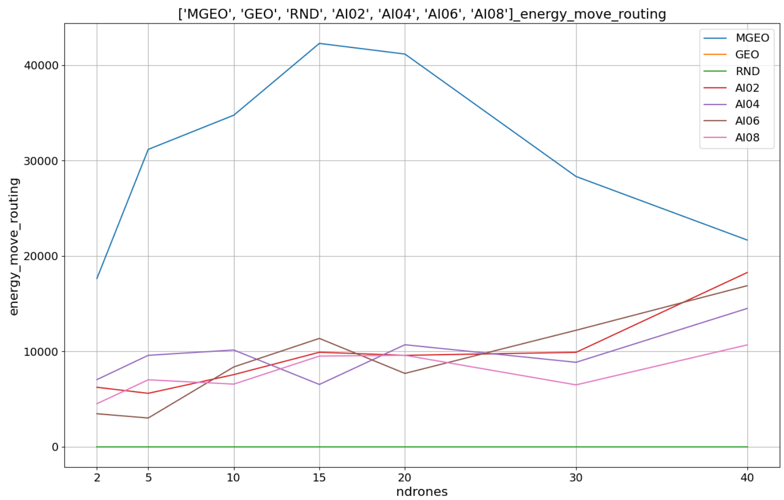

Fig. 3: All α energy consumption

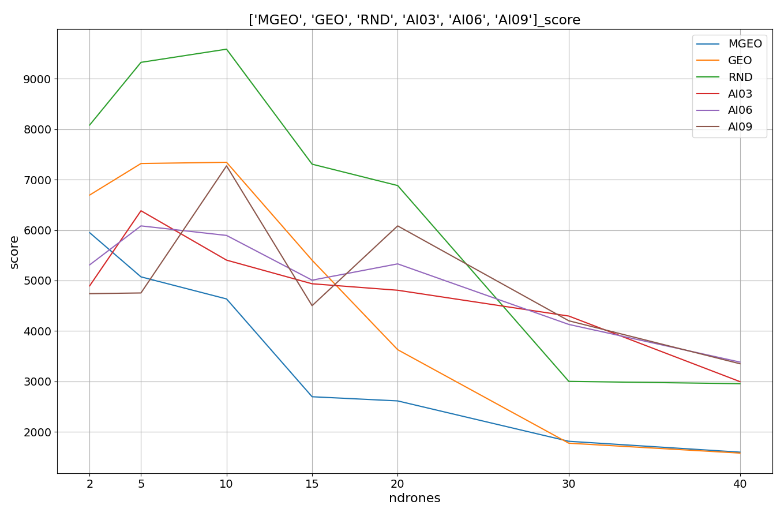

Fig. 4: All γ score

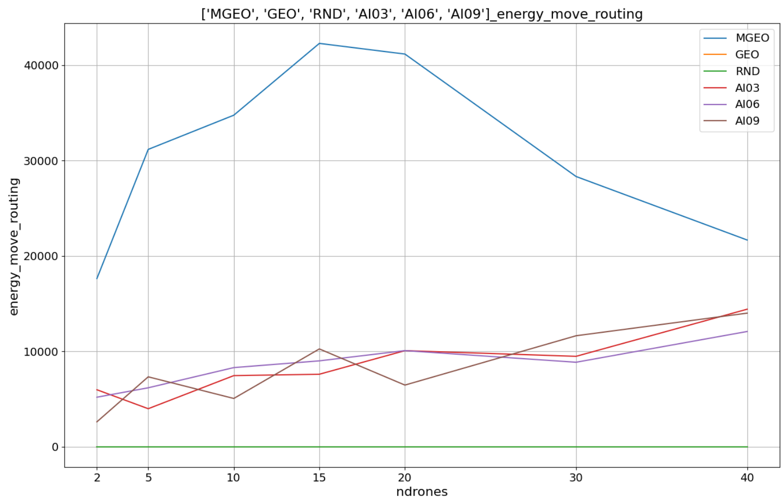

Fig. 5: All γ energy consumption

4 Testing

In this section we show the plots on the performance of the various algorithms with the parameter SWEEP_PATH = True and SWEEP_PATH = False.

4.1 SWEEP_PATH = True

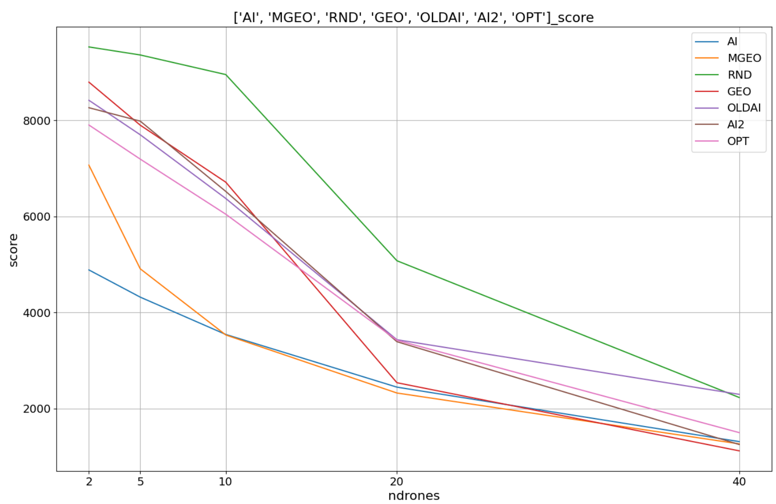

Fig. 6: Score for all algorithms with seed 1 to 30

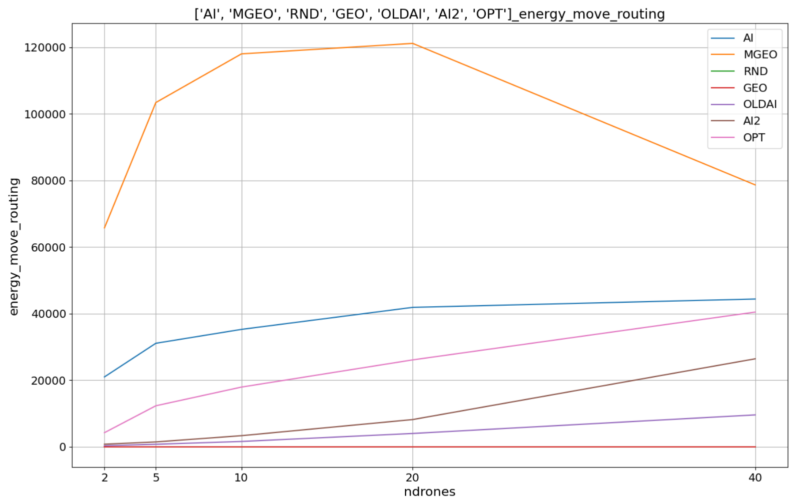

Fig. 7: Energy for all algorithms with seed 1 to 30

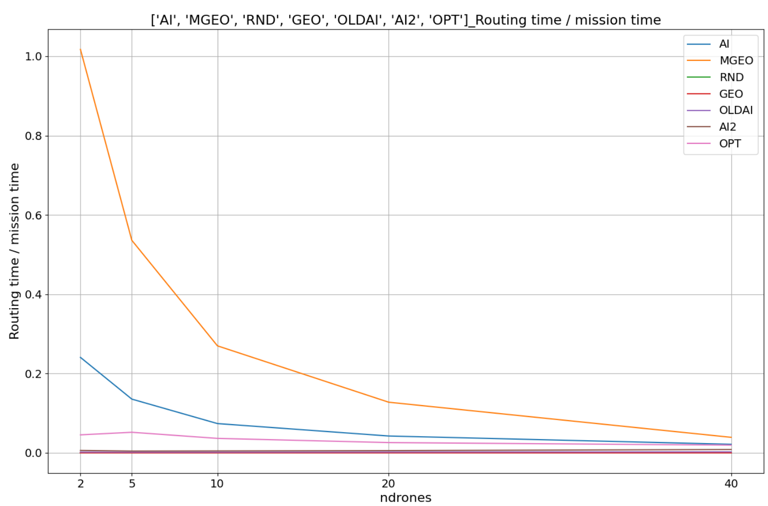

Fig. 8: Mission time for all algorithms with seed 1 to 30

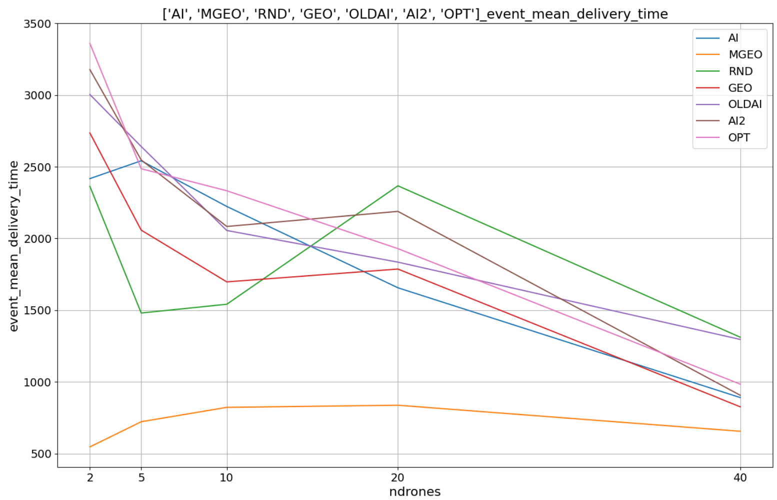

Fig. 9: Mean delivery time for all algorithms with seed 1 to 30

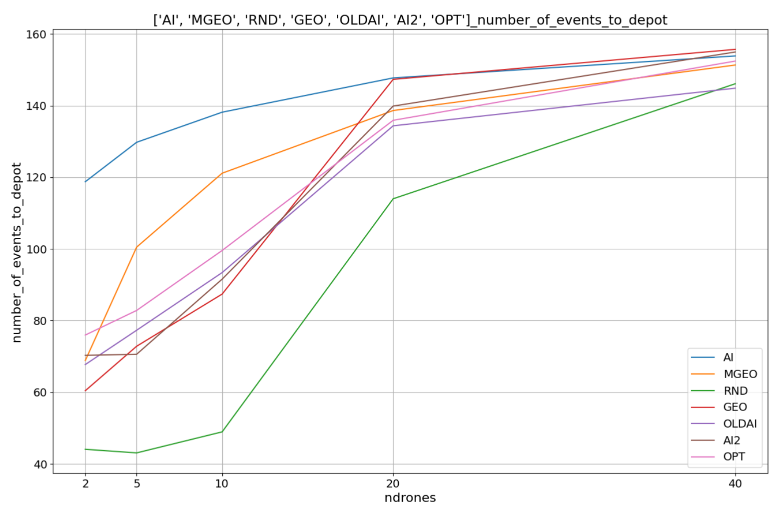

Fig. 10: Number events to depot for all algorithms with seed 1 to 30

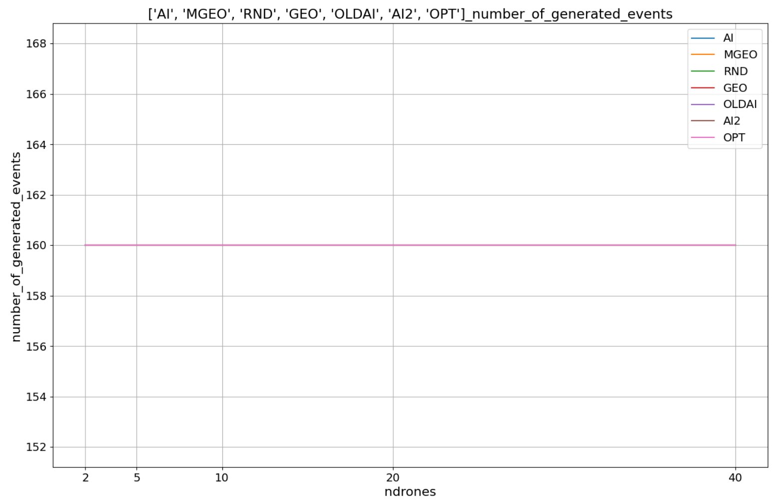

Fig. 11: Number events generated for all algorithms with seed 1 to 30

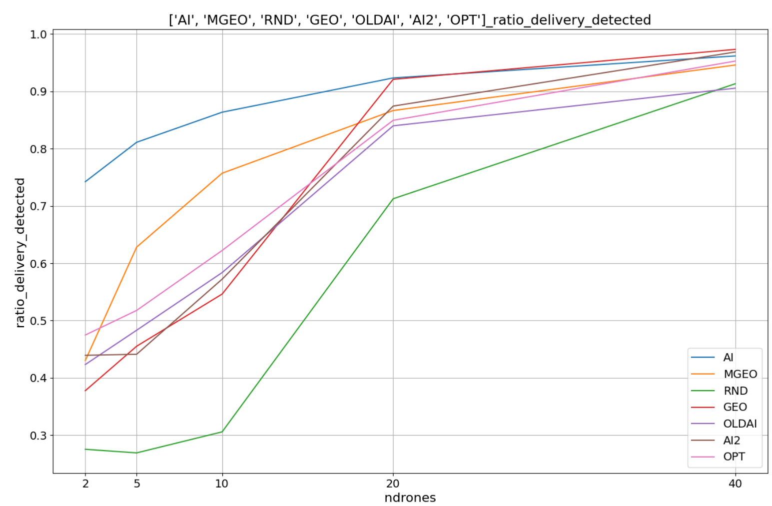

Fig. 12: Ratio delivery detected for all algorithms with seed 1 to 30

4.2 SWEEP_PATH = False

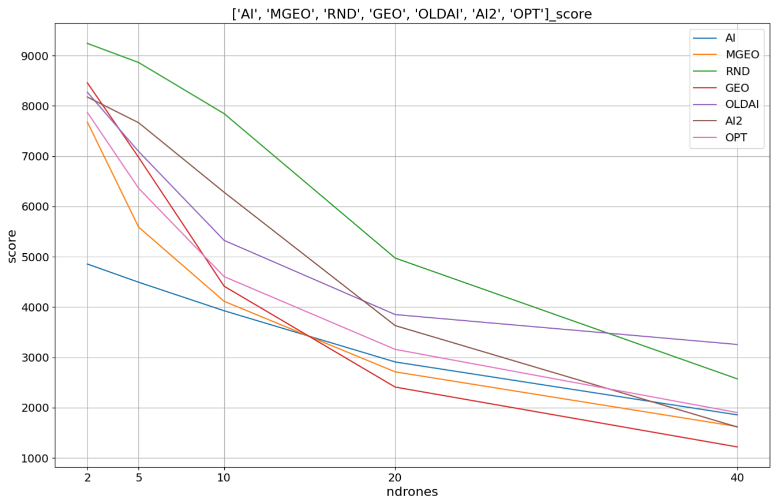

Fig. 13: Score for all algorithms with seed 1 to 30

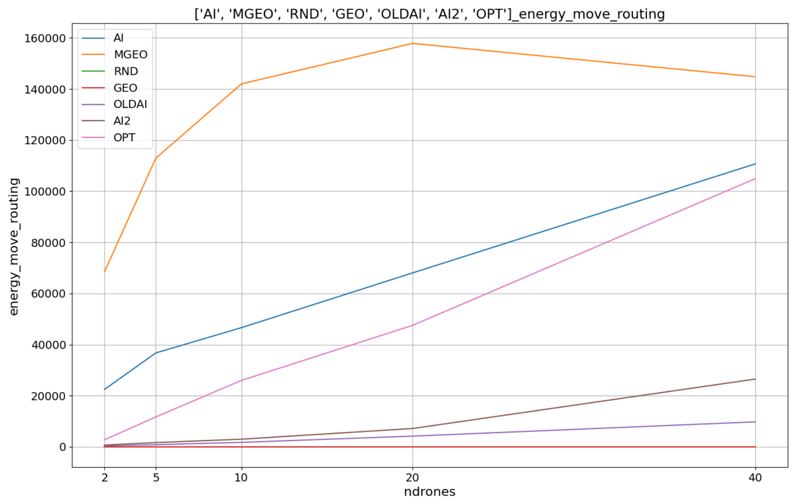

Fig. 14: Energy for all algorithms with seed 1 to 30

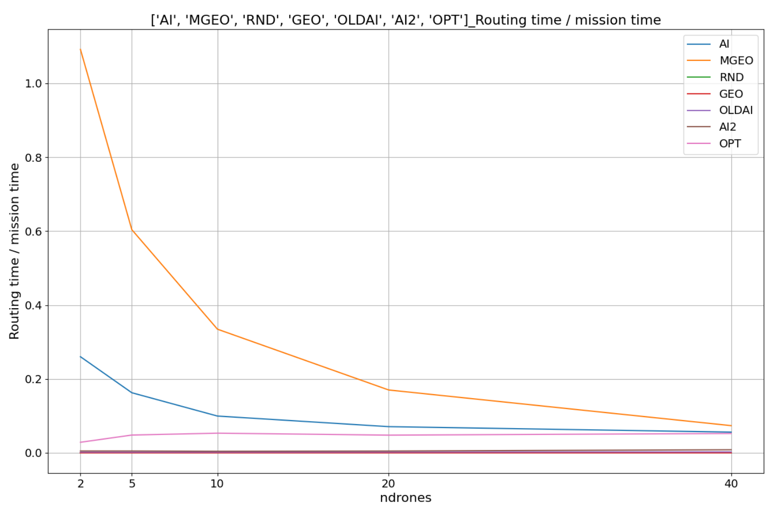

Fig. 15: Mission time for all algorithms with seed 1 to 30

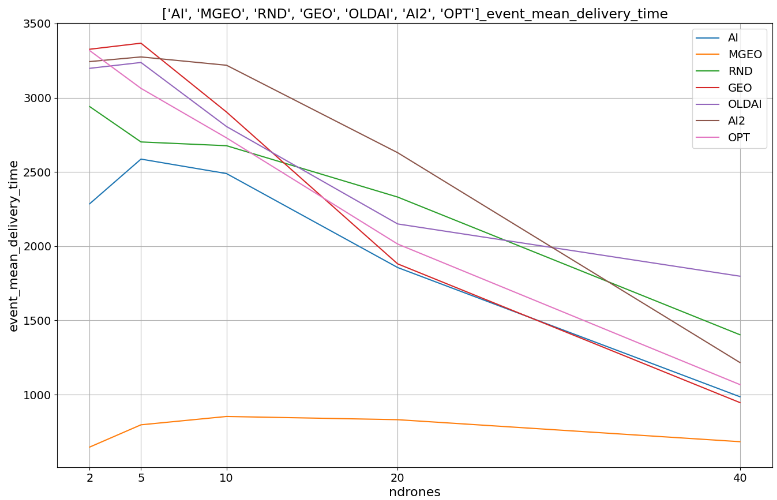

Fig. 16: Mean delivery time for all algorithms with seed 1 to 30

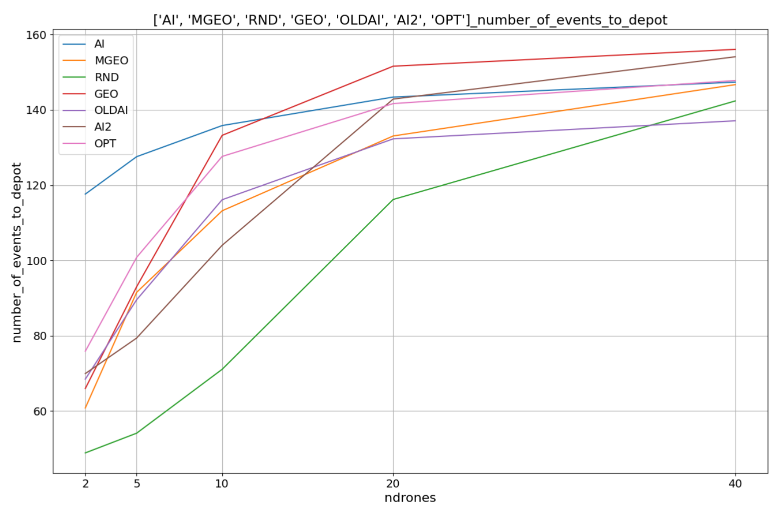

Fig. 17: Number events to depot for all algorithms with seed 1 to 30

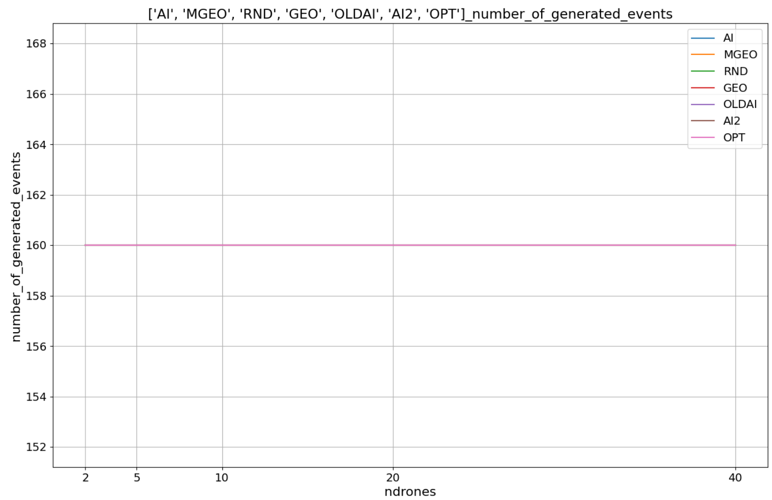

Fig. 18: Number events generated for all algorithms with seed 1 to 30

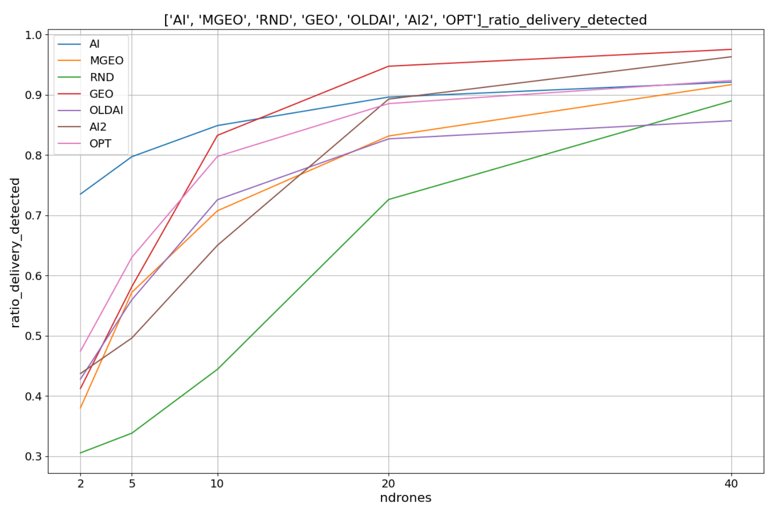

Fig. 19: Ratio delivery detected for all algorithms with seed 1 to 30

5 Conclusions

The approach used to carry out was based on VoIP calls and screen sharing to define the algorithm to use and write an initial draft, discussing the problems encountered and proposing various solutions. Subsequently, each of us made an initial modification of the algorithm by running tests on it, in order to compare ourselves for any changes to be made to the main algorithm. Finally, the testing and plot of the various algorithms used was equally divided to maximize times.

6 Parts developed

6.1 Giovanni Pica

We worked together to the entire project and we divide things such plots, testing or modifying algorithm and in particular I was involved in the development of OLDAI, AI2 and OPT algorithms for compare with our main algorithm.

6.2 Andrea Bernini

We collaborated for most of the project, specifically I was involved in the development and initial testing of the AI algorithm based on the use of the grid and therefore of cells as states.

6.3 Donato Francesco Pio Stanco

We worked together on the whole project, in particular I was involved with the tests on AI and AI2 algorithms. I also took care of the tests and plots for the choice of α and γ.

7 Code of algorithms

- **AI**: `https://pastebin.com/V4rmsvGy`
- **OLDAI**: `https://pastebin.com/7HsuPhVc`
- **AI2**: `https://pastebin.com/yCzvxwa5`
- **OPT**: `https://pastebin.com/pFkJeR7S`