Homework 3 - 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 it is the same as the second homework, with the difference that we have another depot to which we can deliver the packets.

2 Methodology

2.1 Reinforcement Learning Approach

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 ϵ in this homework is divided by the number of drones, because we notice on some plots that with a fixed ϵ the algorithm wastes a big amount of energy. In fact in the ϵ case, of the ϵ-greedy algorithm, the probability of choosing an action that penalizes the energy is 2/4 that is 1/2, compared to 1/3 of the second homework (in this case we have 2 depots instead of one), and with the increase of drones it can become a problem, because the action of going to the depots increases more than in homework 2.

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 down.
- **3**: Move the drone physically to the depot up.

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 (or -2) and then performs the action of going to the depot .

2.2 The state representation

The state representation is the same as the Homework 2 which is based on the Cells.

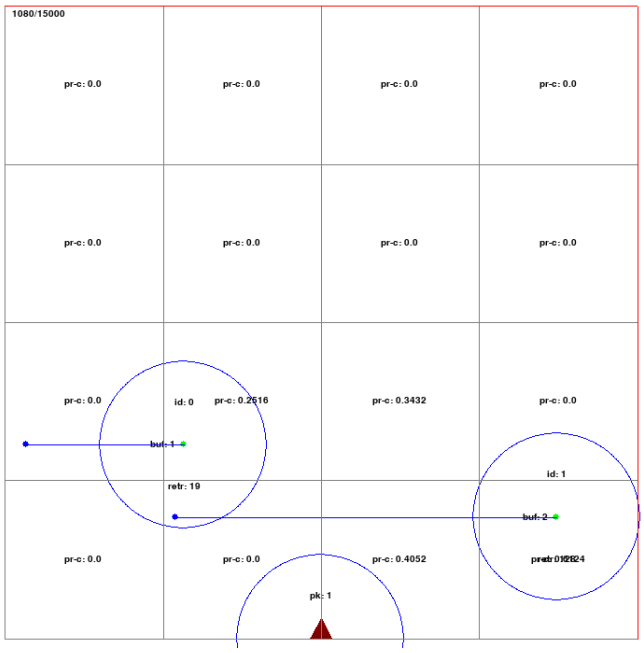

Fig. 1: 4x4 Grid

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 (i.e. the cell) in which the drone is located, as happens in the second homework. So the reward function according to the following scheme:

- **Action 2** (move to depot down) and **Action 3** (move to depot up): The reward decreases as the distance from the depot increases and vice versa. This is because a greater distance from the depot is equivalent to a greater waste of energy.
- **Action 1** (keep the packet):The reward decreases as the distance from the depot increases and vice versa. This is because if the drone is very close it is likely to enter the communication range of the depot. However, unlike the second homework, in this case we have two depots, for this reason we divided the grid in half horizontally, so as to have the cells of the two halves containing mirror values for Action 1. So we will have that the middle row (if the number of rows is odd, rows otherwise) will have the lowest reward values for Action 1.
- **Action 0** (send the packet):The reward increases with increasing distance from the depot and decreases near the depot. This is because it is more convenient to send the package if the drone is in a very distant cell, as the

package is likely to arrive sooner. For the same reason as Action 1 the grid was split horizontally in half with cells containing mirror values for Action 2. So we will have that the middle row (if the number of rows is odd, rows otherwise) will have the highest reward values for Action 0.

Furthermore, when a drone delivers the packages to the depot after performing Action 2 or Action 3, 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.

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. And also for energy consumption is better.

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,
- **OLDAI**: Is the approach used in the first homework for all actions instead of Homework 2.
- **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 ϵ Choice

As a first step we had run several tests on the AI algorithm, to decide which is the best choice for the ϵ value. We have chosen to print epsilon values quite different from each other to have a better view (0.0012, 0.0014, 0.0016, 0.0018). From the graph in Figure 2 it turns out that the best choice is to set epsilon equal to 0.0014.

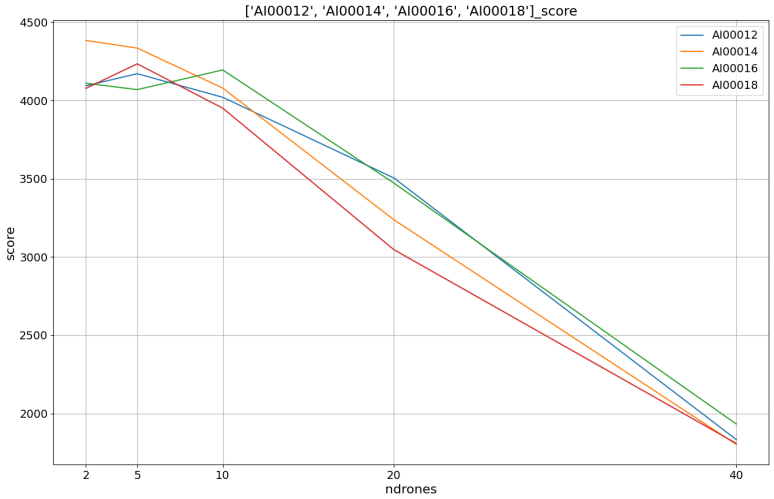

Fig. 2: All ϵ score

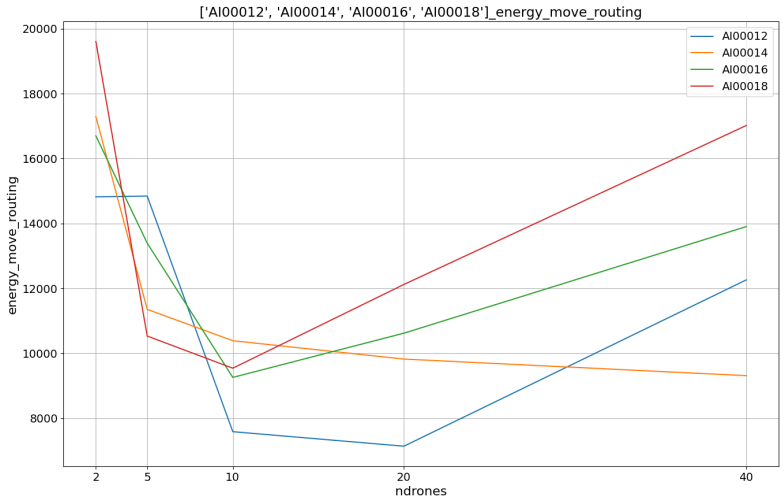

Fig. 3: All ϵ energy

3.3 α and γ Choice

After choosing the ϵ values we applied the same strategy to choose α and γ. For α we have chosen the following values 0.2, 0.4, 0.6, 0.8 and from the graph in Figure 4 it turns out that the best choice is to set α equals to 0.3 because is in the middle of 0.2 and 0.4 also considering the energy consumption. While for γ e have chosen the following values 0.3, 0.6, 0.9 and from the graph in Figure 6 it turns out that the best value for score is 0.3, and the best value for energy consumption is 0.6, for this reason we have chosen 0.5 that is a good compromise between both.

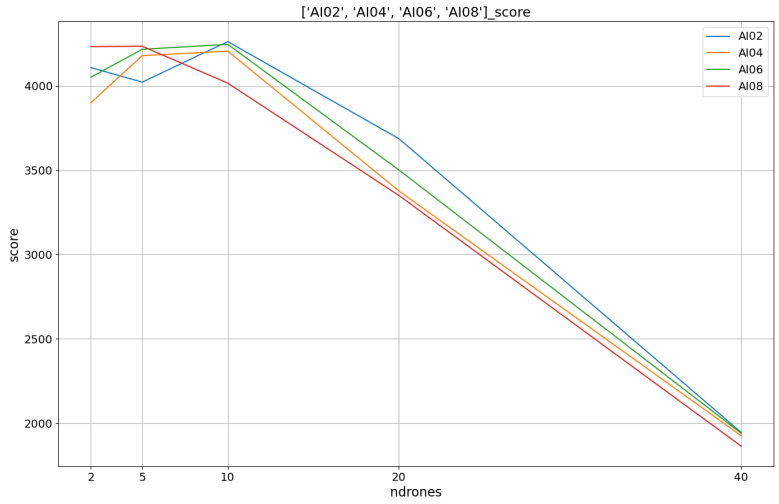

Fig. 4: All α score

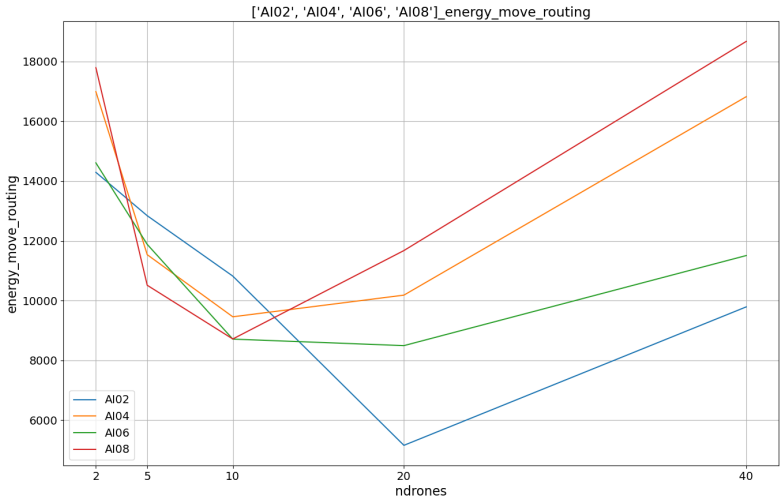

Fig. 5: All α energy consumption

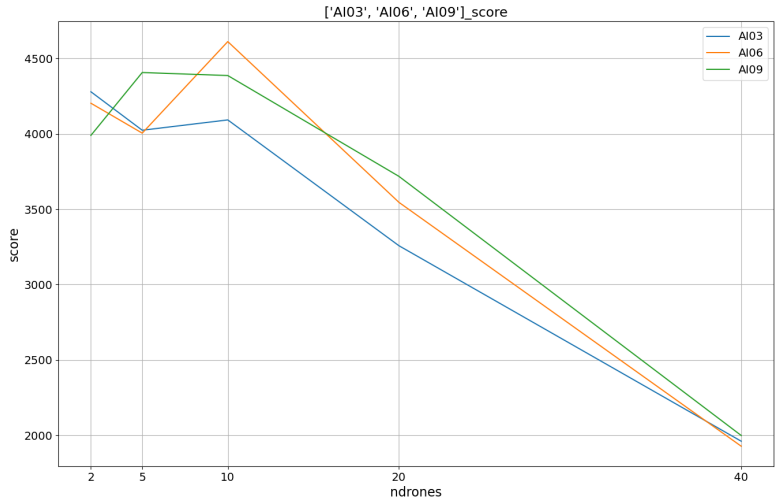

Fig. 6: All γ score

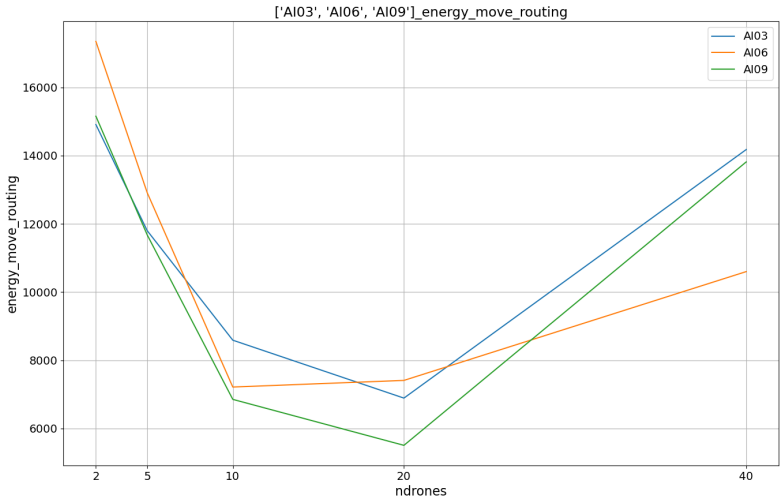

Fig. 7: All γ energy consumption

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

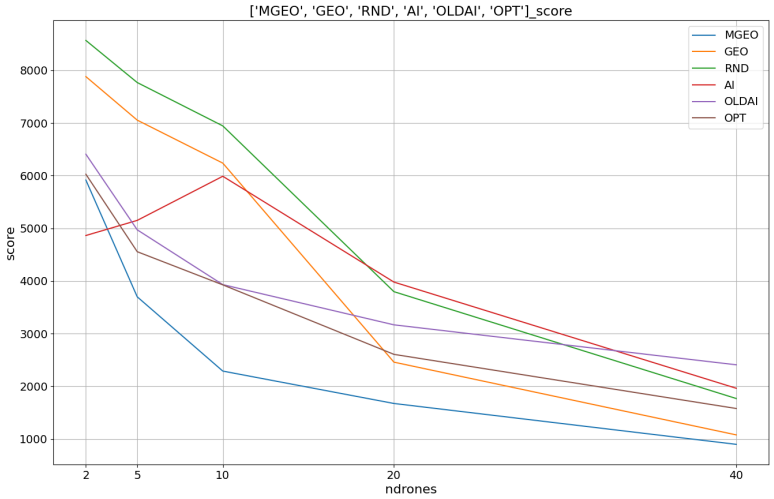

Fig. 8: Score for all algorithms with seed 1 to 30

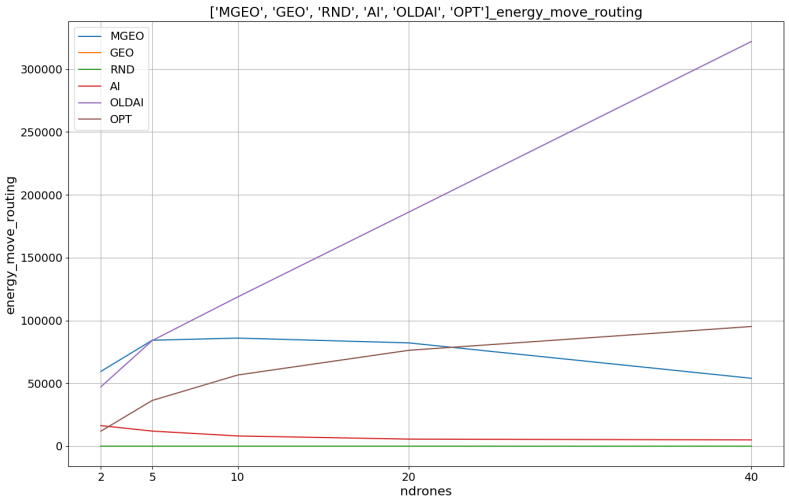

Fig. 9: Energy for all algorithms with seed 1 to 30

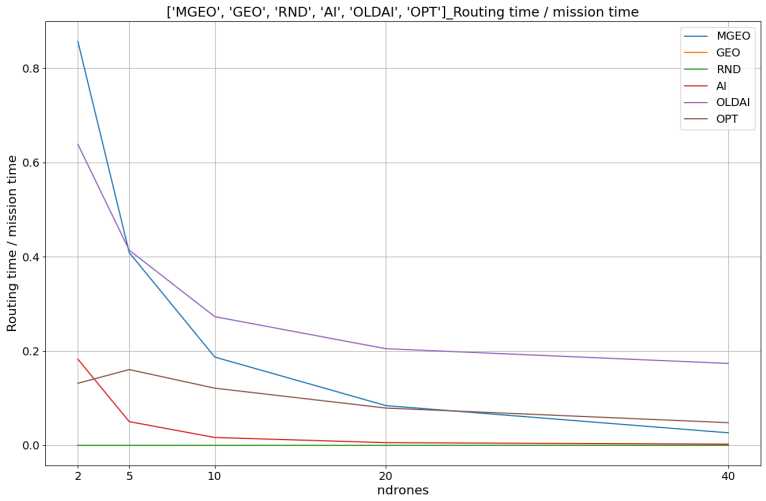

Fig. 10: Mission time for all algorithms with seed 1 to 30

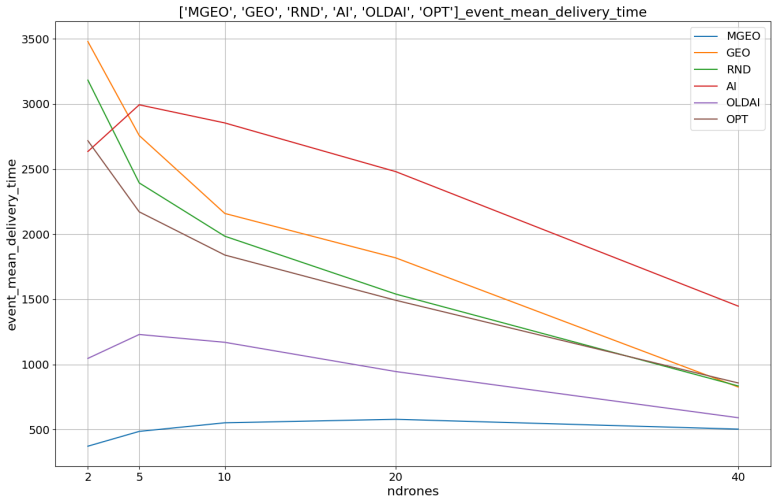

Fig. 11: Mean delivery time for all algorithms with seed 1 to 30

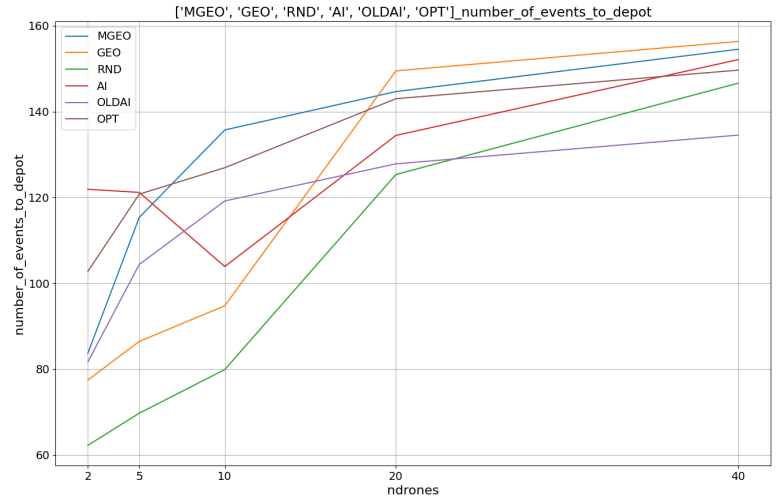

Fig. 12: Number events to depot for all algorithms with seed 1 to 30

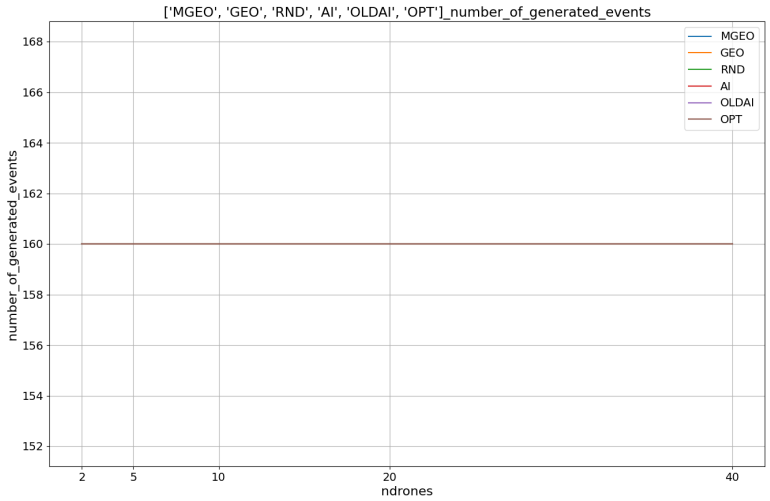

Fig. 13: Number events generated for all algorithms with seed 1 to 30

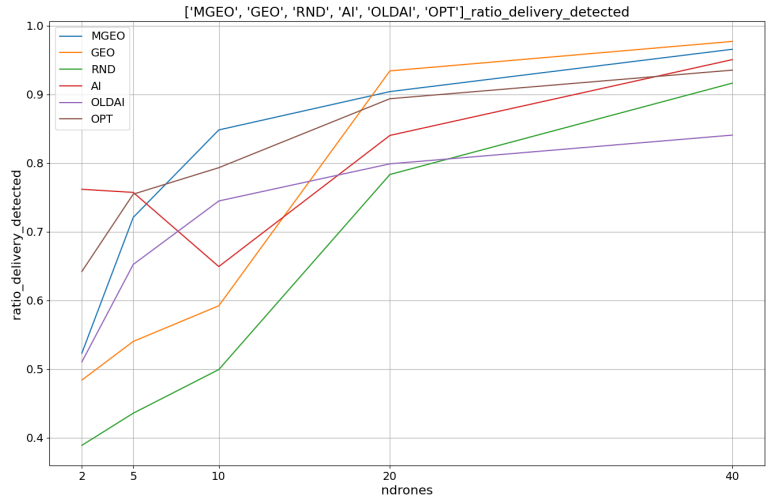

Fig. 14: Ratio delivery detected for all algorithms with seed 1 to 30

4.2 SWEEP_PATH = False

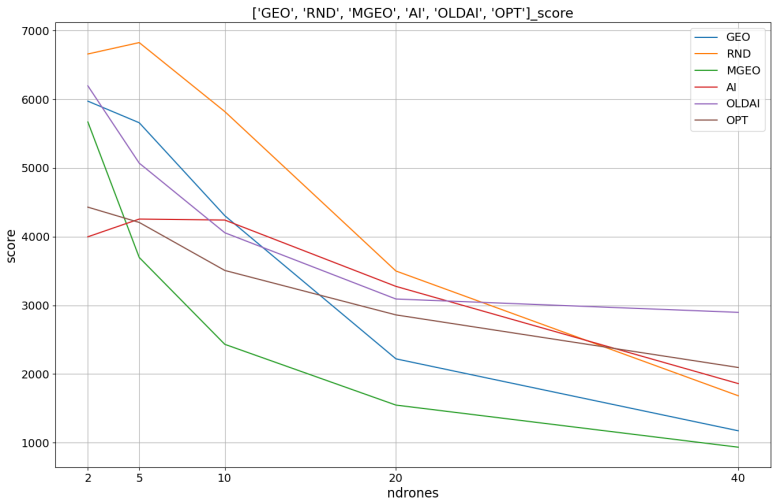

Fig. 15: Score for all algorithms with seed 1 to 30

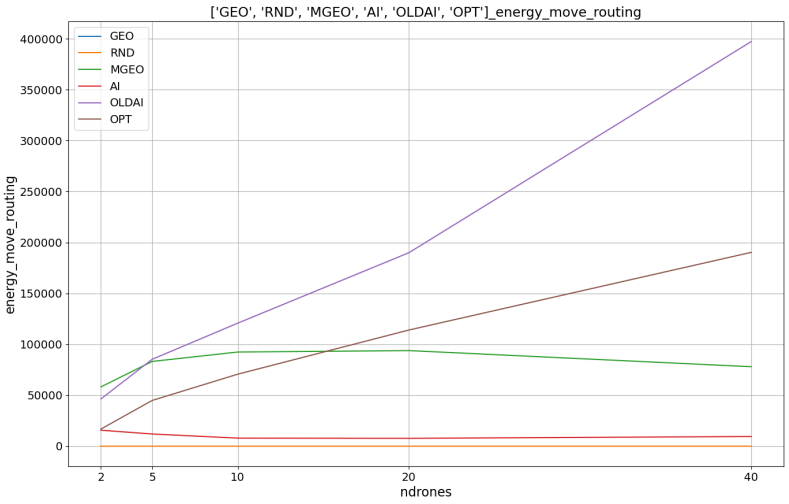

Fig. 16: Energy for all algorithms with seed 1 to 30

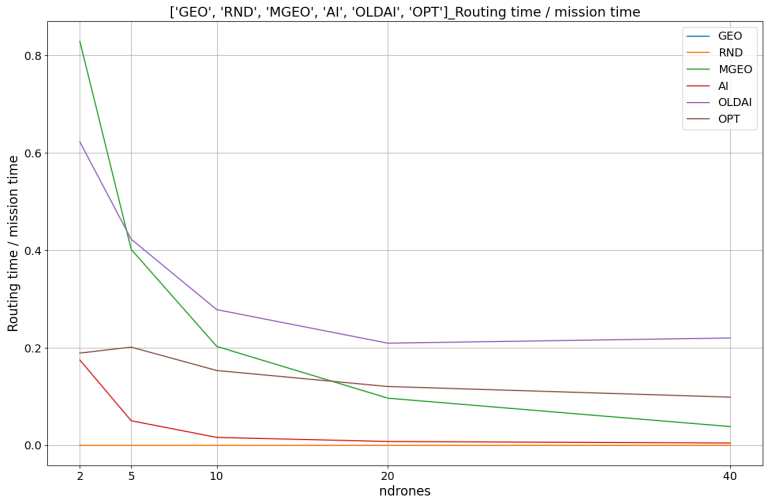

Fig. 17: Mission time for all algorithms with seed 1 to 30

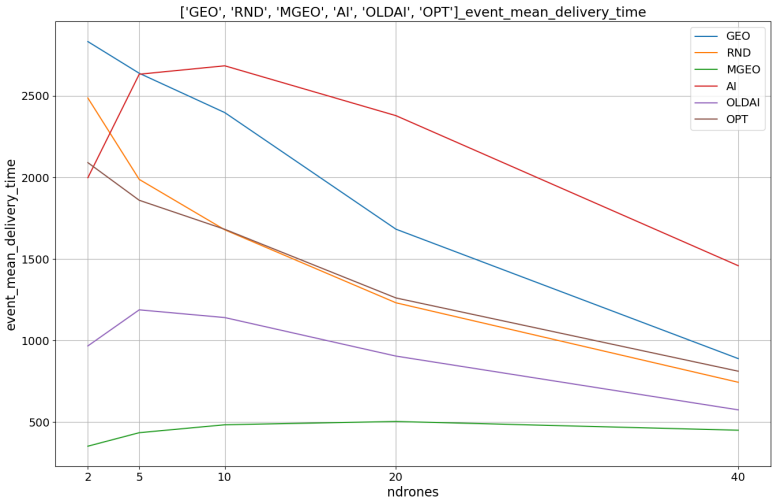

Fig. 18: Mean delivery time for all algorithms with seed 1 to 30

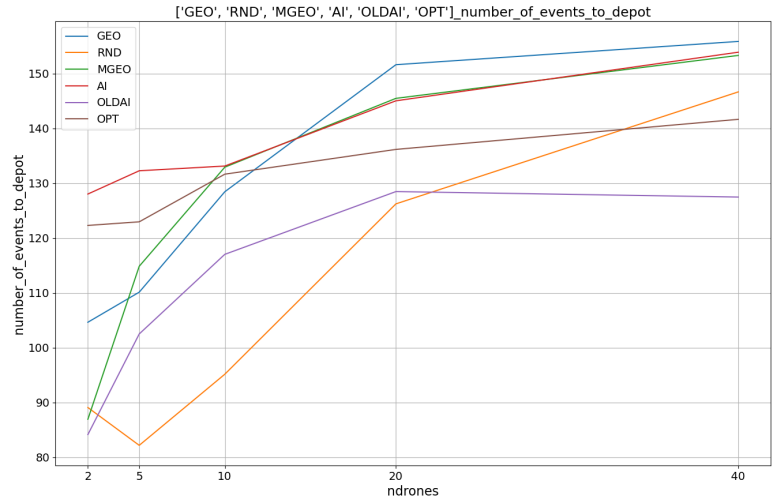

Fig. 19: Number events to depot for all algorithms with seed 1 to 30

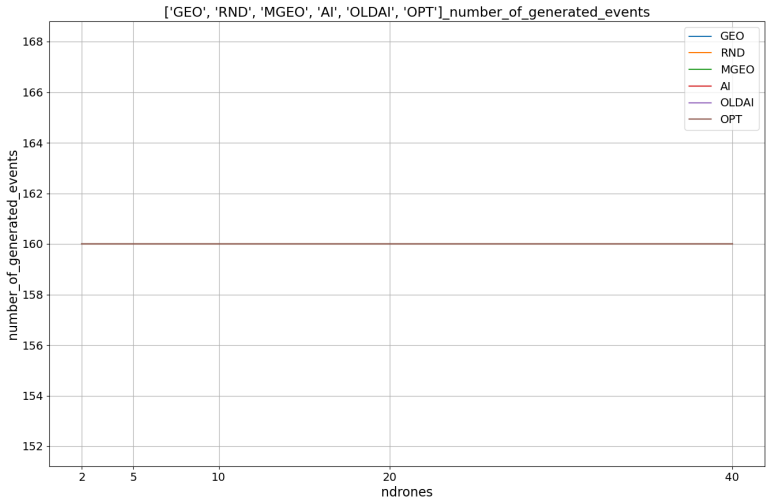

Fig. 20: Number events generated for all algorithms with seed 1 to 30

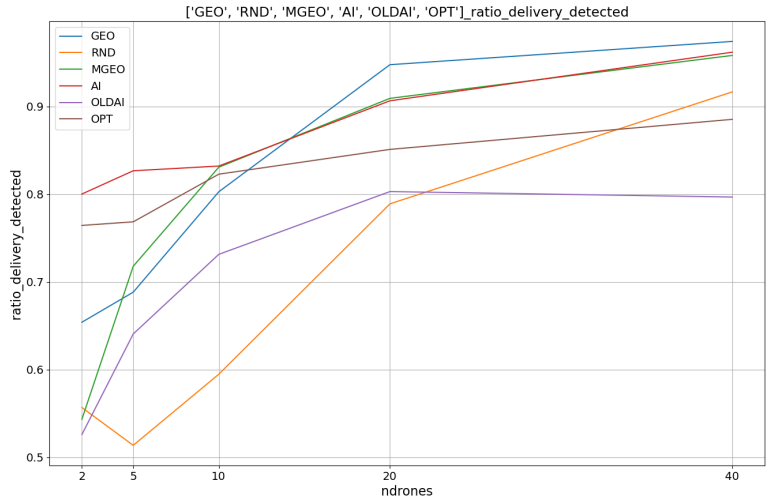

Fig. 21: Ratio delivery detected for all algorithms with seed 1 to 30

5 Conclusions

In this third homework we work together on the developing of this report and also for the modification of the code of the second homework. 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. Finally, the testing and plot of the various algorithms used was equally divided to maximize times.

6 Code of algorithms

- **AI**: `https://pastebin.com/48aSXnW9`
- **OLDAI**: `https://pastebin.com/4avR6Yvq`
- **OPT**: `https://pastebin.com/idpdSKxz`