Artificial Intelligence – Swarm Intelligence

Giulio Turrisi
Course: IA & Robotics

1. Introduction

The final aim of this project, was to coordinate a team of drones to explore one unknown environment, and (separate part), to uproot all the weed present in the field. Speaking about not just a single robot, different approaches was suggested to solve the problem, trying to exploit the collaboration between the different member of the team.
From the three method proposed, I choose the one called “Swarm Intelligence”, which fundamentals derive from the observation of the biological system present in the nature.

This is an approach where the behavior of the single element can be as simple as possible, event without a great computational power. One declination of it, developed in this project, consist in the repetition of the same simple action over and over, without differentiation between the robot.
What is it import is the final aggregation of this simple behaviors, that can create a more complex one useful to complete efficiently the task proposed.

To make this approach more concrete, it’s convenient just to think to a simple example taken from the nature. The one made by the ant! They performs a very simple action(find food for example), but trough the pheromone they can communicate indirectly between each others, starting to create an indirect emerging collaboration!

This kind of communication, it’s called **stigmergy**, and it will be used in this project along another one called “**flocking**”.

2. Stigmergy

Whit this word, we spoke about the totality of the actions through we can communicate between the member of a group, just modifying part of the surrounding environment.

Using the last example about the ant, they can release in the ambient a chemical agent(a pheromone), when they just found a food resource, to inform the other ones about the presence of a point of interest. Following it, they can determine and reach it.

But it’s not only about attractive potential! We can also used it dually, to repulse the team and to misdirect it from one particular location, simple as that.

In our applicative case, we can see this pheromone as an electromagnetic signal, send and captured by an on-board sensor, with which we can maneuver the entire search. We can mark our precedent path to inform the other about the visited zone, and also to signalize the presence of weed in the area.

3. Flocking

The flocking is another basis of the algorithm developed. This is the result of another emergency behavior, that we can see observing the flight of the flock of birds.
They usually tend to move close between each other, trying to use the air flow created to minimize the energy consumed.

There are 3 fundamentals elements when deploying an algorithm using this approach:

- Separation, to avoid imminent impact.
- Alignment, to fly toward the average heading of the neighbors.
- Cohesion, to steer toward the average position of the neighbors.(this is the only one implemented here)

In our case, there is a reason why I choose to combine the use of these two methods. I supposed the presence of concentrate weeds in separate isolated area!
This is why, if our drones use the first approach, they can call the attention of all the near member to make the search for other weed in the area **more efficient and fast**, but this is only feasible if they flight in flock, because just the near ones is supposed to help. (Imagine all the drones in the map leaving their position, wherever they are. It can be quite inefficient.)

4. Goal of the Task and Implementation

Even if the two different task(exploration and weed removal) was proposed as separated one, I tried to implement them together, trying to solve the problem as fast as possible and without an huge increment of the complexity. The method proposed is quite simple to actuate, but not so easy to write on code. One problem of them is **the optimization step.** The are a few variable to setup, as the vision angle(i don't considered all the near robot as flock member, but just the ones in my cone of camera), the attractive-repulsive potentials decade rate and so on.

One way to solve this is through the use a genetic algorithm for example, testing the fitness using different simulation and so on. But it's time consuming and I didn't setup it in time.

The code can be summarize as follow:

1) When the agent need to choose one action, it's start looking around trying to find a neighbor. If there is no one near him(the distance is predefined),or there is but to follow him he need to change too drastically his position(this is done through a "cone of vision" variable), he acts in order to choose the best next action(go in the cell with the best potential). It's a random procedure, with the most feasible cell with an higher chance to be chosen.

2) If somebody is near him and in its cone, he moves toward him. Doing this, he can still choose the best cell in that direction(doing the same as before).

3) If he reaches a wall, he setup an escape move, choosing randomly in all the possible directions. (in the other case, the cell that he can choose are just a sublist of the ones available. This is to reduce the scattered movements that can be difficult to follow if somebody is behind him.)

4) After that, in every cell he goes, he controls the nature of it, and if it is a weed he sprays on that position and inform the other releasing a potential around the zone.

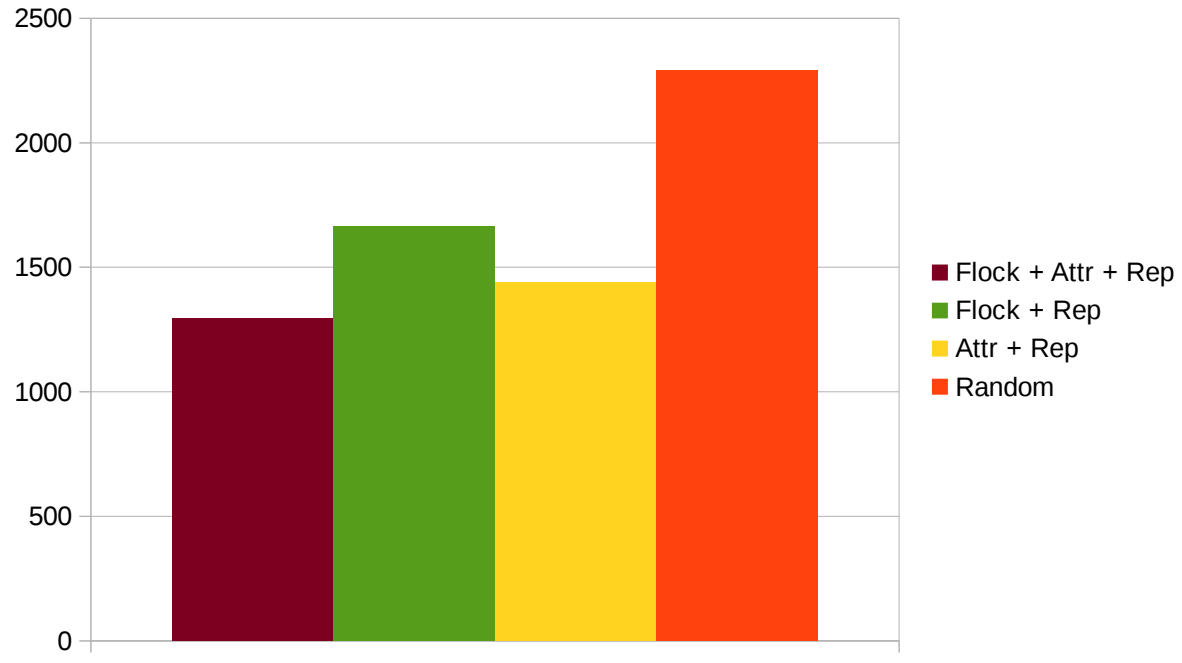
All the potential/information about the weed, is supposed passed by a signal, that can be captured by the other agents in every part of the environment. We can think also to implement a fixed machine who coordinate this kind of communication, but the continuous asking before the choose of one action, can be quite time consuming and slow down all the performance.

5. Final Test and Results

The tests was performed on a grid with 28x28 cell. As I said, the goal was to find and to remove all the weed present in the field, calculating the steps needed to easily compare the approach proposed.

Obviously in a real setup, we can't know if all the weed was removed from the environment, without know the exact number of them. But for test purpose, we can imagine to stop the simulation when this task was completed. But **in the real world**, we can let the robot run for a fixed period of time, being happy to eliminate all the weed we can until the time it's over.

I simulate different number of setups, eliminate part of these methods at a time. With the flock and the potentials activated(attractive and repulsive), with just the potential without the first one, without the repulsive and so on. All these one will be clear just watching the tables below that summarize the runs.



This test was performed until all the weed(less one) was found. It took 20 run each, and this is the average performed on it

How we can see, **the approach proposed** with the cooperation of flock and potential, it's seems a good way to speed up the solution of the problem.

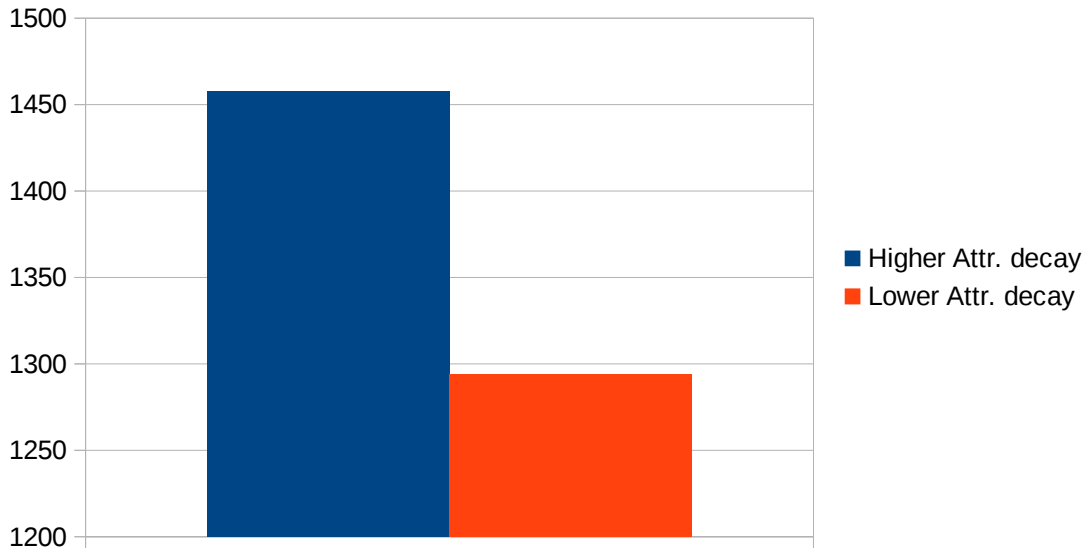
If we drop the attractive potential, the steps needed increase, due to the nature of the environment.

Also the **flock method** seems to play a role on it, and also in this case I think that the map plays an important role in the solution!

If the grid is larger, and the weed remained isolated in little group, the flock will become more and more important to solve easily the task! If we drop the last hypothesis(concentrate weed in-group), all the benefit will became quite lost.
But if we know the structure of our environment, we can switch this different component, making it clever every time and ad-hoc for out setup.

As I said before, a lot of parameters can be optimized(as table show, a quite longer attractive potential seems to make the result worse!), and it can be done with a genetic algorithm and so on.

An example of it, can be found testing a setup with different decay rate of the attractive potential.



If we set it decay too much slow, the drones tend to stay into the area more time, but also if all the weed inside it was found already! If we set it too much hight instead, we can lose the benefit of it, and the drones can go away without explore in depth the area!

In the end, one thing that can be implemented better is the flock part. I just programmed the "cohesion" part of the method, leaving out the others two. A more coordinate and lesser confused formation will be also an help to solve better and better the task assigned.