Ethnocentrism

Agents and Distributed Artificial Intelligence

Ana Santos - up 200700742
Eduardo Ramos - up201505779
Rafael Damasceno - up201504638

2018 / 2019

Introduction

The developed work consists on modelling a population with different races and traits. This model suggests that "ethnocentric" behavior can evolve under a wide variety of conditions. Agents compete for limited space via Prisoner's Dilemma type interactions. The model includes a mechanism for inheritance, both genetic or cultural, of strategies.

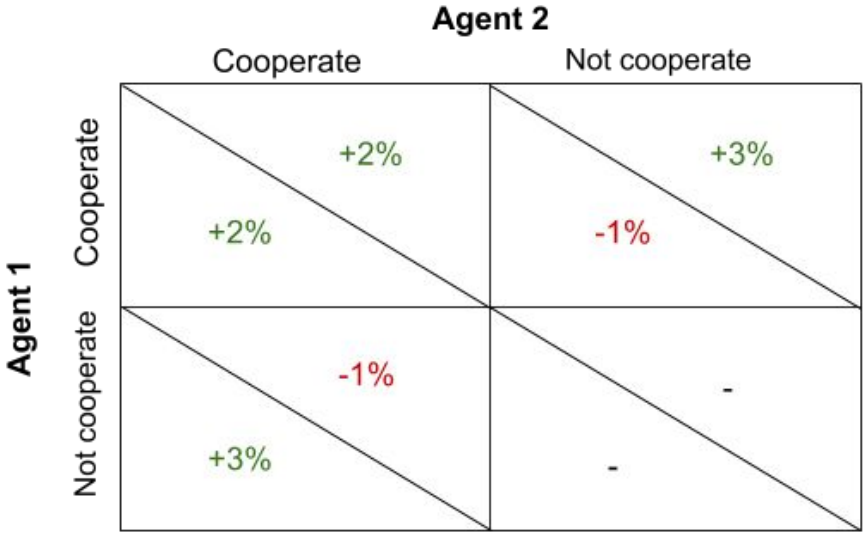


Fig 1 - Prisoner's dilemma example

How it works

Two agent types

World Agent:

- Responsible for the flow of execution;
- Only one per instance
- Creates/Knows all Person agents.

Person agent has three traits:

A. color;
B. whether they cooperate with same colored agents;
C. whether they cooperate with different colored agents.

Resulting agent types :

- An "ethnocentric" only cooperates with same colored agents;
- An "altruistic" cooperates with all agents;
- An ”egoist" cooperates with no one;
- A "cosmopolitan" cooperates with agents of a different color but not of his own.

From the start:

- The world is empty, no agents exist;
- Each new agent that appears has a parameterized chance of cooperating with the same and different color and a constant and equal chance of being from one of any colors;
- Each agent starts its life cycle with a set, parameterized potential to reproduce (PTR)
- Each agents ends its life cycle by being removed from the map.

How it works

The model is divided in cycles during which, the following events occur:

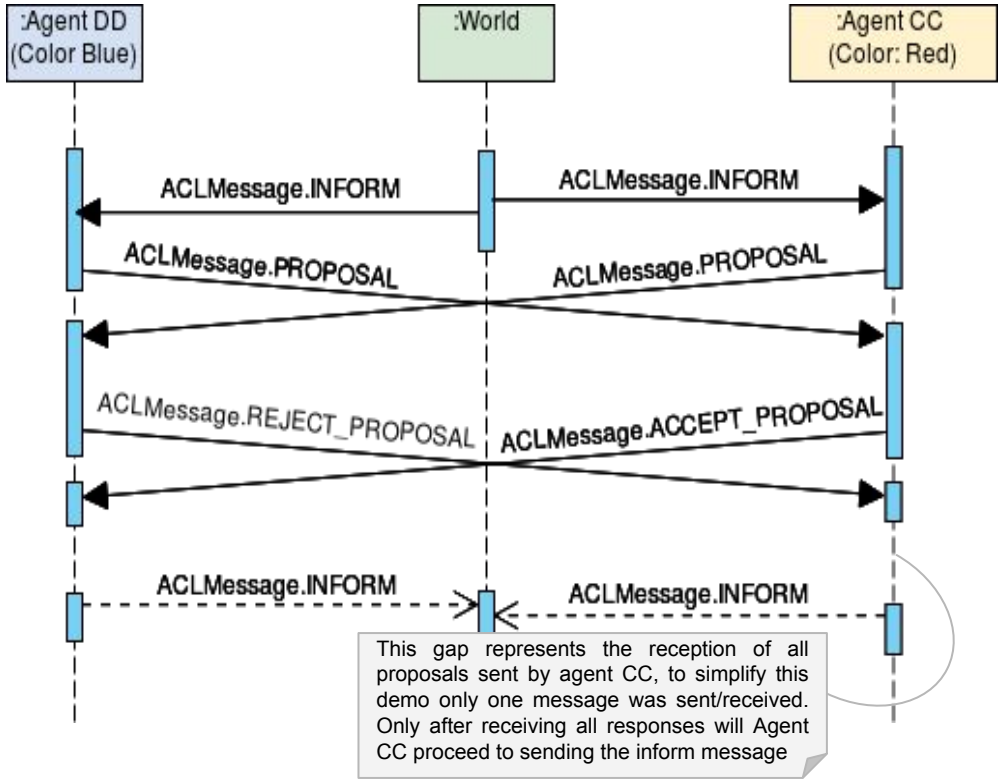
1. The World agent creates and places an immigrant at a random **empty** site on the map.
2. Each pair of adjacent agents interacts in a one-move Prisoner's Dilemma in which each chooses whether or not to help the other. Giving help has a cost, namely a decrease in the agent's PTR by 1%. Receiving help has a benefit, namely an increase in the agent's PTR by 3%.
3. Each agent is chosen in a random order and given a chance to reproduce with probability equal to its PTR. **Reproduction is asexual** and consists of creating an offspring in an adjacent **empty** site, if there is one. An offspring receives the **traits of its parent**, with a **mutation rate** of 0.5% per trait.
4. Each agent has a 10% **chance** of dying, making room for future offspring.

Note: All value with the exception of the increase/decrease of PTR during interaction are parametrized

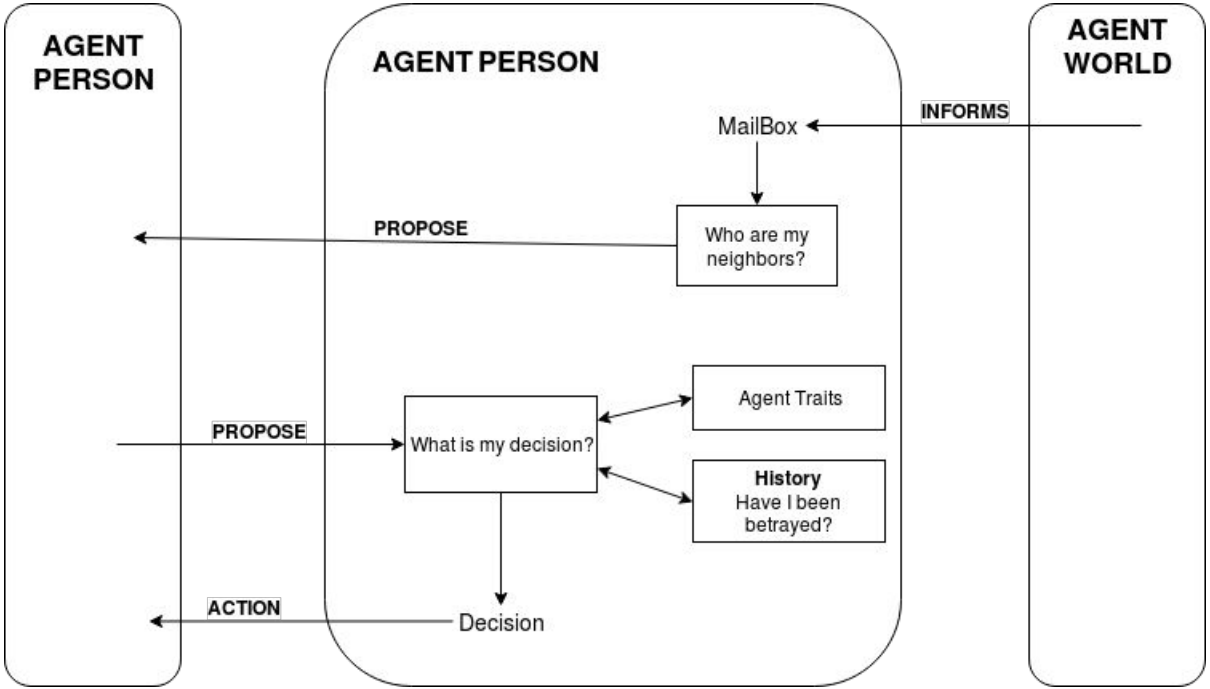
Communication

Communication between the agents was performed through the ACLMessage because it complies with the FIPA 2000 "FIPA ACL Message Structure Specification" specifications. Four types of performative messages were used:

- **ACLMessage.INFORM :** the world agent informs the person agent of its neighbors or the person agent informs the world agent that their interaction with neighbors is over.
- **ACLMessage.PROPOSAL:** the agent makes a proposal of mutual assistance to his neighbors by sending his color.
- **ACLMessage.ACCEPT_PROPOSAL:** the agent receives a confirmation of collaboration.
- **ACLMessage.REJECT_PROPOSAL:** the agent receives a rejection of collaboration.



Architecture of agents



Software

Jade

JAVA **A**gent **DE**velopment Framework is an open source platform for peer-to-peer agent based applications.
This tool was used to ease the development of the agents required for the goal of the project, overall it simplified development by providing all the infrastructure in the format of libraries required to used Agents, ACLMessage communication and Agent Containers.

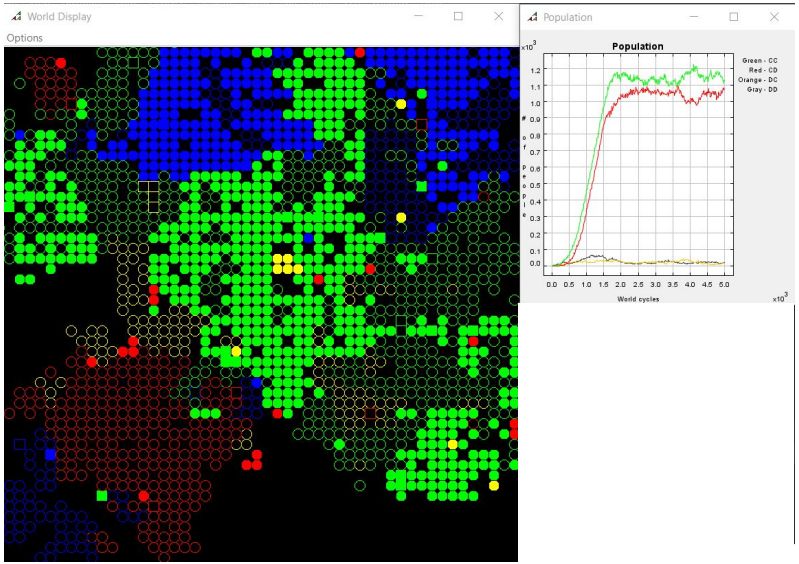
Repast

Recursive **P**orus **A**gent **S**imulation **T**oolkit is a modeling software that allows you to create agent-based simulations focused on modeling social behavior. It allows the scheduling of actions of these agents as well as the construction of a detailed graphical user interface. For this project, only the graphical interface provided by Repast 3 was used. All agent simulation logic was processed using JADE interfaced by SAJaS.

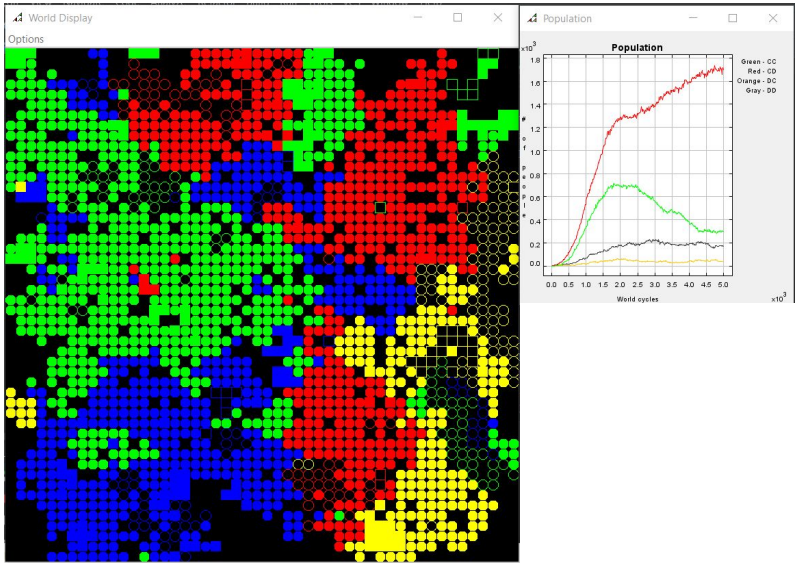
SAJaS

Simple **A**PI for **JA**DE-based **S**imulations is a proposal to bridge the gap between MAS simulation and development. The usage of this tool in the project was to improve performance against JADE based simulation and create the connection necessary to provide a graphical interface.

Experiences



First Experience: Using default parameters and smart choice enabled we can see the better development of the CC population which opts to help only those who will help him back. This prevents this population from losing PTR and lacking space to develop as the CD populations grows faster.
Seed: 1541866056490



Second Experience: Using default parameters and smart choice disabled we can see the better development of the CD population. There is a very noticeable fall in the CC population which can be explained by the loss of PTR is suffers from helping everyone.
Seed: 1541865578164

Results

From the previous experiences, we can see that if the agents are naive and only act as their traits are defined, the CD agents, otherwise known as ethnocentric agents, will slowly overtake the available space in the world, as they help their own race grow without losing PTR to other races, while the CC agents, also known as altruistic agents, lag behind as they lose their own PTR to help everyone. DD agents, or egoist agents, usually are in third, because even though they don't help each other, they also don't hurt their PTR, and DC agents, or cosmopolitan agents, come in last because they don't benefit from helping each other, given they are likely to be born next to agents of the same colour, while constantly sacrificing their own PTR if they interact with agents of different colour.

If the agents stop being naive, meaning that they don't help someone they remember that did not help them, altruistic agents usually come in first because they will reap the most benefits of trying to help everyone, but not losing more PTR than in the first interaction with agents that will not help them. The ethnocentric population will then come in second, with egoist and cosmopolitan populations behaving almost the same, given that cosmopolitans will stop helping those who didn't help them and act almost the same as egoists.

Conclusions

The project successfully implemented the basic ethnocentrism model and allows for the study of the growth of the different kinds of agents, allowing us to see how the limited space and different races influence the growth of a given kind of agent. It also succeeded in removing basic naiveness by simply remembering agents who did not cooperate the first time.

Some interesting work to be done would be to implement models with migrant agents and allowing agents to share the information they have on agents that didn't cooperate with others, thus seeing how well those agents would survive if they knew that no one was helping them in a given area.

Implemented classes

The main implemented classes are:

- Repast3EthnocentrismLauncher
- World
- Person

Other implemented classes are:

- Colour
- WorldState

World

The World class extends Agent and represents a being that controls the world with capabilities of creating and killing persons (inhabitants of the world). As an agent it has a behavior which is a CyclicBehaviour.

This cyclic behaviour is responsible for setting the flow of time and guaranteeing coordination between all agents, a simple state machine was implemented to change state after all agents complete the required actions. There are 4 states: immigration; interaction; reproduction; culling. During immigration the world places a new Person on the map, then Persons on the map interact with their neighbours to vary their PTR. The world is responsible for initiating this phase by sending messages to every person and waiting for all persons to answer before continuing.

After all Persons update their PTR they reproduce, this is an asexual reproduction and as such one single person can reproduce just by having an empty place as one of it’s Von Neumann neighbors.

As a final step each person has a chance to die and the world removes them from existence.

Person

The person class represents an agent that inhabits the world. This agent is able not only to communicate with the world, in order to receive information from the environment that surrounds it, but also with each of its neighbors. As an agent it has a behavior which is a Cyclic Behavior.

This cyclic behaviour is responsible for receiving the messages intended for this agent and for ensuring that each of these messages is answered appropriately. That is, it is the cycle that promotes the agent's decisions:

- If the agent receives a message with neighbor information, these should be contacted with a proposal;
- If the agent receives a message with a proposal it must be processed and can be accepted according to his traits and, if the smart choice is activated, with his interactions' history as well;
- If communication with a neighbors ends, send this information to the World to follow the next process.

Others

Repast3EthnocentrismLauncher:

Class responsible for setting up the world display and plot as well as starting the JADE runtime and creating a main container where all the agents will be placed.

Colour:

Enum used to represent the possible colours a Person can have. This class also provides a method to translate the colour to a java.awt.color class to be used with repast.

WorldState:

Enum used to represent the possible world states.