Titolo Progetto (da decidere)

Luca Rickler, Kim Nicoli, Francesco Parino

Abstract

Lo mettiamo?

Contents

1 Introduction

1.1 General Overview

The subject we present in this paper, has originally been inspired by the Kolkata Paise Restaurant problem; more in detail we took inspiration from the paper **"The Kolkata Paise Restaurant Problem and Resource Utilization" from Chakrabarti et al, March 2009.**

The problem mentioned in the paper is modelled as a One-Shot game which takes place in Kolkata, the Indian capital of West Bengal, where there are many cheap and fixed rate restaurants usually frequented by workers at lunch time.

As a matter ~~of fact, in~~ order to save the transport costs, they usually walk down to one of these cheap restaurants to have lunch during their lunch break. So, here is the problem: every single restaurant's capacity to serve customers is limited, and every diner will not serve more than a fixed number of customers on each day. As a matter of fact, if most of the workers go to the same restaurant, this, will not be able to satisfy every ~~single customer~~ and some ~~of them~~ will not get lunch and consequently will go back to work hungry and unsatisfied.

...satisfy every one of them and some will not...

Moreover, if someone goes to an already filled restaurant, then, walking down to another restaurant would mean failing to report back to work in time. It is even important to underline that in the classic version of the problem there is the assumption that every restaurant serves lunches of different qualities - according to a ranking, known to the customers - but that the prices are the same anywhere, no matter how high the quality of the restaurant is.

The problem is quite simple to understand, but, on the other hand, is quite difficult to be solved. It is obvious to imagine that every worker possibily wants to go to the best ranked restaurant in Kolkata and be served. In light of what we saw before, this is unrealistic because the capacity of every restaurant is limited; maybe one restaurant can serve more than one customer each day, but will not be able to satisfy all the workers of the city.

The Kolkata Paise Restaurant problem can be seen as a complication of the El Farol bar problem, a problem in game-theory based on a bar in Santa Fe, New Mexico, created in 1994 but, in fact, formulated and solved six years earlier, in 1988, with a different name. In this **paper**, we will face the problem by using the agent-based modelling technique.

1.2 ~~Motivation:~~ solving the problem through a MAS model

This problem has already been modelled by an Agent Based Model (ABM) project, written in Netlogo, where smart agents tried to find the best strategy in order to maximize their own satisfaction, without comunication with other people.

Now, our aim is to bring the problem on an upper level by modelling a similar situation through a Multi Agent System (MAS) project, which is quite different. Where we had smart agents acting on their own, now we have agents who interact among them by the introduction of a mutual communication. Although agents are not as intelligent as they were in the ABM model, now they can communicate and bring out a collective intelligence through out their social behaviour, led by their interaction.

The main target of this work is to outline how the communication among agent at a very low level could influence the rise of a collective behaviour.

It is even important to underline that the problem studied by Chakrabarti (*Chakrabarti et al., March 2009*) was a One-Shot Game. Hence, we had *One-Shot behaviours* which means that people just act once in looking for an available restaurant[1], if they fail (don't find a good restaurant) they immediately stop.

Our situation is far more different; we have to deal with *cyclic behaviours*. Every agent has a protocol to follow and a list of task predicates to satisfy; thus, if he does not find a restaurant at the first attempt he tries again with another one, until he find a place with available tables or he realises, after having checked all restaurants, that there are no tables available anywhere so that the task can not be satisfied.

2 Model Design

2.1 Agent types and hierarchy

We have basically two type of agents: People and Restaurants.

People are more dynamical agents since they take decisions, update their kwoledge database continuously and contribute to modify the external environment.

Moreover, they hold many different characteristics.
boldness, a *minimum evaluation* and a ~~*maximum number*~~ list of friends *of friends*.

- "*Boldness*" referres to the attitude of the agent to learn from what he experienced. That means if he had a bad experience in a certain restaurant if his boldness is low he would not return there again, while, if is boldness is high, he would probably give to the restaurant a second chance.
- The *minimum evaluation* referres to the lowest restaurant quality a person is minded to accept. This means that an high value of this parameter represents a very exigent customer while a low one will lead the agent anywhere, no matter the quality of the food.
- The last parameter, is the ~~*maximum number*~~ list of friends *of friends* that every person has; these friends could influence the thought of a single agent by communicating him their thinking. Every agent, for each one of his friends, has an attribute which is the "*trust*" that this person put in all of them; we could say, in this sense, that my best friend is the one I believe the most.

Nuovamente, le scale sono invertite, minimo = buono, massimo = cattivo

La mappa di thrust contiene tutte le persone, non solo gli amici

On the other hand, restaurants are more simple than people. As a matter of fact we have only 2 parameters to deal with: the capacity and the rank.

[1] in the paper authors assumed that every restaurant had just one place available.

- As we could expect a single restaurant could not serve everyone; that's the reason why we need a *capacity*. Once a restaurant is filled it can not accept other bookings. So people who expected to go there need to change their goal.

- The second parameter, *the rank*, outlines the objective quality of a restaurant. We fixed an evaluate range from 0 to 5, where five means worst quality restaurants while zero are the best ones.

2.2 Model Protocol

The structure of the communication is quite simple. As mentioned before we treat the problem differently from the **paper**. In our analysis we have to deal with a Repeated-Game instead of a One-Shot Game (which is far more abridging).

Qui usiamo il termine protocollo diversamente da come lo uso poi nella parte di implementazione

We organised our protocol in turns. For each turn, every single person follows a precise sequence of actions:

- Search: first of all, an agent needs to find among all restaurants the one which is the best for him.

- Choose: secondly he must control that this restaurant has available places; thus, if the best restaurant is already full he goes for the second and so on.

- Eat: then, once he found a good restaurant to book at, he goes there and eats.

- Evaluate: finally, the agent updates his own opinion of the restaurant, for the next turn, grounding on what he experienced.

This protocol will be explaind in a more detailed way later in section (INSERIRE SEZIONE IN CUI SI SPIEGA LA COMUNICAZIONE).

2.3 Strategies

The action of choosing the best restaurant is an interesting and important thing to analyse. In this sense we select three main strategies that an agent can follow in choosing the best place to eat. It is important to underline that the probability a single customer chooses one strategy instead of another can be set before launching the simulation.

The strategies we consider are:

- Rational Choose: this is the most common way of choosing. The customer selects all restaurants that satisfies his *minimum evaluation*, and among them chooses the best one which has available places.

- Irrational Choose: the second strategy brings to our model some fuzzy logic through an irrational way of choosing. As a matter of fact an agent chooses at random, among the restaurants that have available places, where to go.
- Best Friend Choose: the third strategy is probably the most interesting to outline the influence of the communication between people in our model. In this strategy a person decides to go to the restaurant which is thought the best one by the friend in which he believes the most (his *best friend*!). Obviously, if the restaurant is already full, the agent will go for the second preference of his best friend "top ranked restaurants" and so on, until he finds the best restaurant with available places.

3 Model Implementation

Da qui si riaggancia luca