Learning Nash Equilibria in Non-Cooperative Games

Alfredo Garro
University of Calabria, Italy

INTRODUCTION

Game Theory (Von Neumann & Morgenstern, 1944) is a branch of applied mathematics and economics that studies situations (games) where self-interested interacting players act for maximizing their returns; therefore, the return of each player depends on his behaviour and on the behaviours of the other players. Game Theory, which plays an important role in the social and political sciences, has recently drawn attention in new academic fields which go from algorithmic mechanism design to cybernetics. However, a fundamental problem to solve for effectively applying Game Theory in real word applications is the definition of well-founded solution concepts of a game and the design of efficient algorithms for their computation.

A widely accepted solution concept of a game in which any cooperation among the players must be self-enforcing (non-cooperative game) is represented by the Nash Equilibrium. In particular, a Nash Equilibrium is a set of strategies, one for each player of the game, such that no player can benefit by changing his strategy unilaterally, i.e. while the other players keep their strategies unchanged (Nash, 1951). The problem of computing Nash Equilibria in non-cooperative games is considered one of the most important open problem in Complexity Theory (Papadimitriou, 2001). Daskalakis, Goldbergy, and Papadimitriou (2005), showed that the problem of computing a Nash equilibrium in a game with four or more players is complete for the complexity class PPAD-Polynomial Parity Argument Directed version (Papadimitriou, 1991), moreover, Chen and Deng extended this result for 2-player games (Chen & Deng, 2005). However, even in the two players case, the best algorithm known has an exponential worst-case running time (Savani & von Stengel, 2004); furthermore, if the computation of equilibria with simple additional properties is required, the problem immediately becomes NP-hard (Bonifaci, Di Iorio, & Laura, 2005) (Conitzer & Sandholm, 2003) (Gilboa & Zemel, 1989) (Gottlob, Greco, & Scarcello, 2003).

Motivated by these results, recent studies have dealt with the problem of efficiently computing Nash Equilibria by exploiting approaches based on the concepts of learning and evolution (Fudenberg & Levine, 1998) (Maynard Smith, 1982). In these approaches the Nash Equilibria of a game are not statically computed but are the result of the evolution of a system composed by agents playing the game. In particular, each agent after different rounds will learn to play a strategy that, under the hypothesis of agent's rationality, will be one of the Nash equilibria of the game (Benaim & Hirsch, 1999) (Carmel & Markovitch, 1996).

This article presents SALENE, a Multi-Agent System (MAS) for learning Nash Equilibria in non-cooperative games, which is based on the above mentioned concepts.

BACKGROUND

An n-person strategic game G can be defined as a tuple $G = (N; (A^i)_{i\in N}; (r^i)_{i\in N})$, where $N = \{1, 2, \dots , n\}$ is the set of players, A^i is a finite set of actions for player $i \in N$, and $r^i : A^1 \times \dots \times A^n \rightarrow \Re$ is the payoff function of player i. The set A^i is called also the set of pure strategies of player i. The Cartesian product $\times_{i\ N} A^i = A^1 \times \dots \times A^n$ can be denoted by A and $r : A \rightarrow \Re^N$ can denote the vector valued function whose ith component is r^i, i.e., $r(a) = (r^1(a), \dots , r^n(a))$, so it is possible to write (N, A, r) for short for $(N; (A^i)_{i\in N}; (r^i)_{i\in N})$.

For any finite set A^i the set of all probability distributions on A^i can be denoted by $\Delta(A^i)$. An element $\sigma^i \in \Delta(A^i)$ is a mixed strategy for player i.

A (Nash) equilibrium of a strategic game $G = (N, A, r)$ is an N-tuple of (mixed) strategies $\sigma = (\sigma^i)_{i\in N}$, $\sigma^i \in \Delta(A^i)$, such that for every $i \in N$ and any other strategy of player i, $\tau^i \in \Delta(A^i)$, $r^i(\tau^i,\sigma^{-i}) \leq r^i(\sigma^i,\sigma^{-i})$, where r^i denotes also the expected payoff to player i in the mixed extension of the game and σ^{-i} represents the mixed strategies in σ of all the other players. Basically, supposing that all the other players do not change their strategies it is not possible for any player i to play a different strategy τ^i able to gain a better payoff of that gained by playing σ^i. σ^i is called a Nash equilibrium strategy for player i.

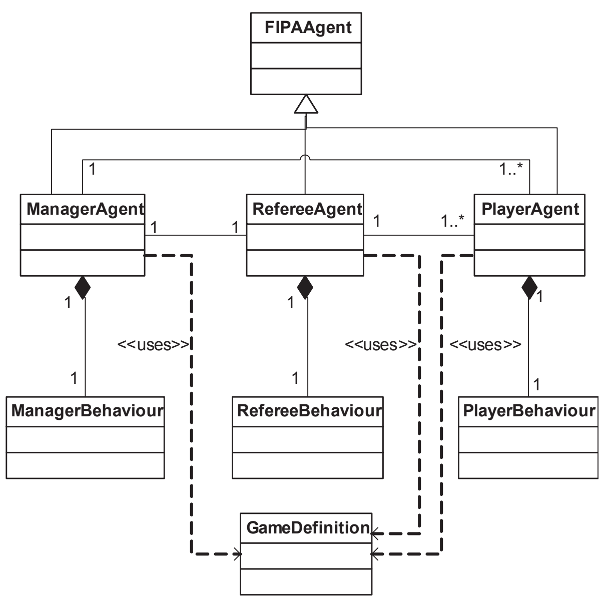
In 1951 J. F. Nash proved that a strategic (non-cooperative) game $G = (N, A, r)$ has at least a (Nash) equilibrium σ (Nash, 1951); in his honour, the computational problem of finding such equilibria is known as NASH (Papadimitriou, 1994).

SOFTWARE AGENTS FOR LEARNING NASH EQUILIBRIA

SALENE was conceived as a system for learning at least one Nash Equilibrium of a non-cooperative game given in the form $G = (N; (A^i)_{i\in N}; (r^i)_{i\in N})$. In particular, the system asks the user for:

- the number n of the players which defines the set of players $N = \{1, 2, \ldots, n\}$;
- for each player $i\in N$, the related finite set of pure strategies A^i and his payoff function $r^i : A^1 \times \ldots \times A^n \rightarrow \Re$;
- the number k of times the players will play the game.

Figure 1. The class diagram of SALENE



Then, the system creates n agents, one associated to each player, and a referee. The agents will play the game G k times, after each match, each agent will decide the strategy to play in the next match to maximise his expected utility on the basis of his beliefs about the strategies that the other agents are adopting. By analyzing the behaviour of each agent in all the k matches of the game, SALENE presents to the user an estimate of a Nash Equilibrium of the game. The Agent paradigm has represented a "natural" way of modelling and implementing the proposed solution as it is characterized by several interacting autonomous entities (players) which try to achieve their goals (consisting in maximising their returns).

The class diagram of SALENE is shown in Figure 1.

The Manager Agent interacts with the user and it is responsible for the global behaviour of the system. In particular, after having obtained from the user the input parameters G and k, the Manager Agent creates both n Player Agents and a Referee Agent that coordinates and monitors the behaviours of the players. The Manager Agent sends to all the agents the definition G of the game then he asks the Referee Agent to orchestrate k matches of the game G. In each match, the Referee Agent asks each Player Agent which pure strategy he has decided to play, then, after having acquired the strategies from all players, the Referee Agent communicates to each Player Agent both the strategies played and the payoffs gained by all players. After playing k matches of the game G the Referee Agent communicates all the data about the played matches to the Manager Agent which analyses it and properly presents the obtained results to the user.

A Player Agent is a rational player that, given the game definition G, acts to maximise his expected utility in each single match of G without considering the overall utility that he could obtain in a set of matches. In particular the behaviour of the Player Agent i can be described by the following main steps:

1. In the first match the Player Agent i chooses to play a pure strategy randomly generated considering all the pure strategies playable with the same probability: if $|A^i|=m$ the probability of choosing a pure strategy $s\in A^i$ is $1/m$;
2. The Player Agent i waits for the Referee Agent to ask him which strategy he wants to play, then he communicates to the Referee Agent the chosen

Related Content