A REINFORCEMENT LEARNING APPROACH FOR MULTIAGENT NAVIGATION

Francisco Martinez-Gil, Fernando Barber, Miguel Lozano, Francisco Grimaldo
Departament d'Informatica, Universitat de Valencia, Campus de Burjassot, Burjassot, Valencia, Spain
Francisco.Martinez-Gil@uv.es, Fernando.Barber@uv.es, Miguel.Lozano@uv.es, Francisco.Grimaldo@uv.es

Fernando Fernandez
Departamento de Informatica, Universidad Carlos III, Campus de Leganes, Madrid, Spain
fernand@inf.uc3m.es

Keywords: Reinforcement learning, Multiagent systems, Local navigation.

Abstract: This paper presents a Q-Learning-based multiagent system oriented to provide navigation skills to simulation agents in virtual environments. We focus on learning local navigation behaviours from the interactions with other agents and the environment. We adopt an environment-independent state space representation to provide the required scalability of such kind of systems. In this way, we evaluate whether the learned action-value functions can be transferred to other agents to increase the size of the group without loosing behavioural quality. We explain the learning process defined and the the results of the collective behaviours obtained in a well-known experiment in multiagent navigation: the exit of a place through a door.

1 INTRODUCTION

During last years, the most popular approaches to multiagent navigation have been inspired in different kind of rules (phyisical, social, etological, etc). These systems have demostrated that it is possible to group and to combine different rules (eg. cohesion, obstacle avoidance (Reynolds, 1987)) to finally display high quality collective navigational behaviours (eg. flocking). However the main drawbacks are also known. All the rules must be defined and adjusted manually by the engineer or author. The number of rules required for modelling complex autonomous behaviours can be high, and generally they are systems difficult to adjust when scaling the number of agents, where some problems (eg. local minimum or deadlocks) can appear. Beyond handwritten rules systems, other discrete techniques, as celular automata or multi-layer grids have been also used in these domains (Lozano et al., 2008) for representing different kind of navigational maps, as they can precompute important information for the agents when they have to plan and to follow their paths.

In this paper we present a Reinforcement Learning (RL) approach that consists on modelling the problem as a sequential decision problem using Markov Decision Process (MDP). Reinforcement learning techniques have been used successfully to find policies for local motion behaviors without knowledge of the environment (Kaelbling et al., 1996). It has been also applied in cooperative tasks, where several agents maximizes the collective performance by maximizing their individual rewards (Fernández et al., 2005). In these cooperative tasks, like Keepaway (Stone et al., 2005), the relationship among individual rewards and the cooperative behavior is typically unknown, but colaboration emerges from the individual behaviors.

The aim of the paper can be summarized in: a) setting a RL multiagent local navigation problem to simulate a single-door evacuation and b) to study the possibility of transferring the learned behaviors from a specific scenario to other bigger and more populated environments, which represents the basic concept of scalability in this domain. We propose the use of multiagent reinforcement learning using independent learners to avoid the "curse of dimensionality" of pure multiagent systems. We do not model the problem as a single-agent RL problem to allow the emergence of different solutions providing variability to the simulation. We explore the scalability (to hundreds of agents) and portability (to larger grids) of the approach. Scalability from a reduced set of agents to large ones is performed through the transfer of the value functions (Taylor and Stone, 2005). We

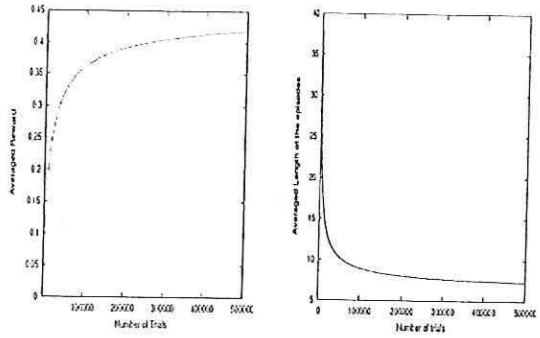

Figure 2: Mean reward curve and mean length of the episodes curve for one agent in the learning process.

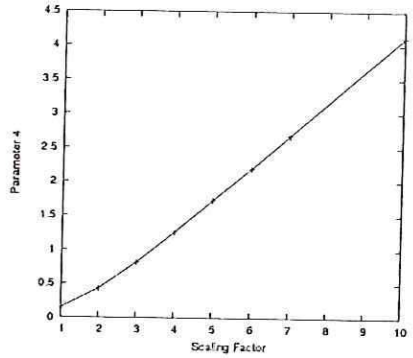

Figure 3: Parameter 4 for the experiment. These data are averages over 100 trials

3 SCALING UP THE NUMBER OF AGENTS

In simulation, we will use the learned value functions in a larger environment with different number of agents. We exploit the action-value functions using the greedy action selection as the optimal policy for an agent. Since the action and the state spaces are the same that used for learning, no mapping is required. However a generalization of the distance is necessary because the new grids are bigger in simulation time and agents can be placed initially farther than the learned distances. Our generalization criteria is based on the idea that the distance feature loses its discriminatory power when it is large. Therefore the distances higher than the $MaxDist - 1$ value are mapped to a distance in the range $[MaxDist - 1, MaxDist/2]$ using an empirical criteria.

Each action-value function is used to animate an incremental number of agents in the simulation environment to know their scalability. Thus then, the number of simulated agents grows in a factor $\times 1$, $\times 2$, $\times 3$,...,$\times 10$ with the following meaning: a set of 20 functions corresponding with the learning process of 20 agents will have a scaling sequence of $\times 1 = 20$ agents, $\times 2 = 40$ agents, $\times 3 = 60$ agents, etc.

4 EVALUATING THE EXPERIMENT

We have defined five evaluation parameters.

1. Parameter 1. It is the mean of random actions carried out by an agent and it is related with the quality of the learned action-value function. When the generalization process described formerly fails, the agent chooses a random action.
2. Parameter 2. It is the total number of crashes that happened in the simulation time. Bad learning of states is a possible situation because convergence is warranted only with infinite iterations and to visit infinitelly often all the states is not guaranteed due to the on-line data adquisition based in the interaction with the environment.
3. Parameter 3. It is the total number of simulated episodes that have ended without success. When the agent has spent a maximum number of steps in one episode, it finishes with no success.
4. Parameter 4. It is the relative difference between the minimum amount of steps necessary to arrive to the exit from the initial position and the actual number of steps used. It represents the value $(\frac{l_{act}-l_{min}}{l_{min}})$ where l_{min} is the minimum number of actions to reach the exit from a position with a single agent and l_{act} is the number of actions actually carried out. It gives the idea of the difference between the actual performance and the minimum possible number of actions to reach the door. A value of 1.0 means that the number of actions is two times the minimum.
5. Parameter 5. It is an average density map that let us to estimate the collective behaviour achieved by the multiagent system during simulation. It gives a shape of the crowd in the grid, that is a reference parameter normally considered in pedestrian dynamic simulations (Helbing et al., 2000).

We have performed scaling simulations up to a scaling factor of $\times 10$ in the number of agents, corresponding to a maximun of 200 agents.

Concerning the Parameter 1, the percentage of aleatory actions used in simulation is 0% for our experiment in all the scaling factors. This result shows two facts: the generalization strategy has provided candidates in all the cases and the Q function for states