Learning Social Behaviors without Sensing

Anand Panangadan **Michael G. Dyer** *
Computer Science Department
University of California, Los Angeles
Los Angeles, California 90095
{anand, dyer}@cs.ucla.edu

Abstract

A learning algorithm is presented that enables agents that do not have the ability to sense other agents to adapt its behaviors (that were learned in a single agent environment) to novel situations (deadlocks arising from existing in an autonomous multi-agent system). This adaptation takes place as the agent continues to perform its construction task. When the agents are confined to narrow spaces, this learned behavior leads to a "bucket brigade". The algorithm also learns the pattern of activations on its spatial map that is associated with deadlocks and the new behaviors are exhibited when this pattern is later observed.

1 Introduction

(Chao et al., 2000) described a behavior-based architecture with connectionist action selection that enabled an agent to rearrange objects in its continuous two-dimensional simulated world into a pre-specified pattern. In this work, a learning mechanism is introduced that enables this architecture to be used in the multi-agent scenario *without* extending the sensory capabilities of the agents. Agents cannot detect other agents and deadlocks can arise between two agents with interfering paths. The learning algorithm enables the agents to learn to drop any "brick" being carried in case of a deadlock so that the other agent can pick it up and replan its path. The learning is unsupervised as an agent uses the progress it has made since trying out an action as reinforcement.

2 Environment and Architecture

The simulated environment is 2-dimensional and continuous. Distance sensors detect bricks but not other agents. If an agent tries to move to a location occupied by another, it will not succeed. Agents can pick up a brick (carrying it as it moves) and drop the brick. Construction in this world thus involves moving toward a brick, grabbing it, moving to one of the specified drop sites and then dropping the brick at that location.

*This work supported in part by an Intel University Research Program grant to the second author.

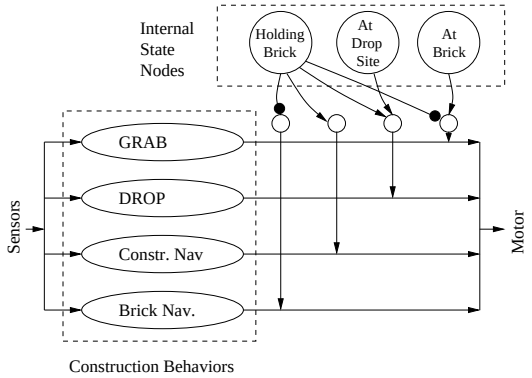

Figure 1: Construction behaviors and internal state nodes

Egocentric Spatial Maps (ESMs) are used to represent the location of bricks around an agent. An ESM is a grid of neurons that divides the area around the agent into small uniform squares such that the central neuron represents the square on which the agent is currently present. Behaviors plan paths to bricks and drop sites by spreading activation on the ESMs. The output of the behaviors is sent to the connectionist action selection module (figure 1) which chooses one of these behaviors depending on the current stage of the construction task (encoded by *Internal State* nodes).

3 Learning Social Behaviors

To recognize deadlocks involving other agents, a new internal state node, f, that measures the "frustration" of the agent is added along with weighted links to behaviors. If the agent is unable to move in a timestep, the activation on this node increases. When this activation exceeds a threshold, θ, then the agent tries to perform a random behavior (grab, drop, move toward brick or drop-site). If the agent is able to get out of the deadlock, then the weights are changed to reinforce this behavior. If the agent remains deadlocked, then that behavior is penalized. Since the action selection network is a single layer network, the Perceptron learning rule is used:

1: At time t, if ($a_f > \theta$) then
2: with probability p_o perform an action o

3: **At time** $(t+\delta)$, **if** $|\vec{p}(t) - \vec{p}(t+\delta)| > d$
4: **then** $\Delta\ w_{io} = \text{sign}(0.5 - a_o)\eta\ a_i\ e^{-kw_{io}^2}$
5: **else** $\Delta\ w_{io} = \text{sign}(a_o - 0.5)\eta\ a_i\ e^{-kw_{io}^2}$

where a_i denotes the activation of internal state node i. a_o denotes the activation of output node o (representing some behavior), w_{io} the weight on the link between i and o, and η is the learning rate. p_o in step 2, reduces the probability that both agents are taking random actions at the same time. The test whether the random behavior was successful is carried out at time $t+\delta$ by checking if the change in position $\vec{p}$ is over some distance d. The exponential term is present to bound the weights w_{ij} and to reduce the rate at which large weights (learned previously) change.

These learned behaviors are triggered only when the agent is caught in a deadlock. However, if deadlocks tend to occur in narrow passageways, then the same set of ESM neurons will be activated during deadlocks (since ESMs represent space egocentrically). A simple update rule is used to learn these map activations, activated each time the agent is in a deadlock:

$$l_i \leftarrow l_i + \eta_m(s_i - l_i)e^{-kl_i{}^2}$$

where s_i is the activation of neuron i, l_i is the corresponding learned value, η_m is the learning rate and the exponential term is used to retain values learned from previous time steps. The agent now activates its learned behaviors whenever its ESM activations match the learned weights.

4 Results

Two learning agents: The agents retain the normal construction sequence when f is inactive. When f is active and an agent is holding a brick, it drops it even if it is not at a drop site (if the agent is not holding a brick, no behavior is selected). Learning occurred faster with increasing p_o but behaviors that did not contribute to breaking the deadlock were also reinforced.

Five learning agents: The weights learned were similar to that of the two agent case in environments with open spaces. If there are narrow passageways, more than two agents are often involved in deadlocks and since the time to break such deadlocks will depend on the number of agents, learning occurs slower than in open spaces.

A learner and a previously trained agent: The learner learns to drop its brick when frustrated and holding a brick, but continues to attempt the brick navigation behavior when not holding a brick. This is because the trained agent immediately performs the drop behavior when it is holding a brick, and does not give the learning agent an opportunity to explore its choice of behaviors.

Bucket Brigade: Five agents (trained in pairs), were placed in an environment where bricks and drop sites were separated by a long corridor. When all agents are within this corridor, a bucket brigade is formed through a sequence of pair-wise interactions between agents.

Map association: Two learners were placed in an environment with a narrow corridor. Most deadlocks occurred within this corridor and the agents associated activations on neurons close to the center of the ESM with deadlocks. Agents then drops discs when these neurons are activated (in any corridor) even with no deadlock.

5 Conclusions and Related Work

The adaptation took place while the agents continued to perform their construction task. The mechanism does not require external supervision as it utilizes the feedback provided by the environment. The bucket brigade behavior was learned from purely local interactions.

Learning was faster when there were two agents learning simultaneously compared to the case when one agent was already trained. Also, in the experiments with the learned ESM activations, only one of the agents could use the learned activations (as otherwise both agents would try to drop discs within corridors). The issue of heterogeneity was not studied here and such conditions were explicitly satisified by the experimenters.

(Ostergaard et al., 2001) studies the performance of the "bucket brigade" behavior in different environments. The use of "frustration" to trigger learning is similar in spirit to the impasse driven learning in Soar (Laird et al., 1987) and to the use of progress estimators to speed up reinforcement learning (Mataric, 1994). The architecture described here is novel in its use of a connectionist action selection mechanism that enables simple learning rules to adapt its behaviors, goals and spatial representation. This connectionist approach can be compared to the action selection mechanism in (Maes, 1991) that connects behaviors with activation spreading links.

References

Chao, G., Panangadan, A., and Dyer, M. G. (2000). Learning to integrate reactive and planning behaviors for construction. In *Proc. of the 6th Intl. Conf. on the Simulation Of Adaptive Behavior.*

Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, 33:1–64.

Maes, P. (1991). A bottom-up mechanism for action selection in an artificial creature. In *Proc. of the First Intl. Conf. on Simulation of Adaptive Behavior.*

Mataric, M. J. (1994). Reward functions for accelerated learning. In *Machine Learning: Proc. of the Eleventh Intl. Conf.*

Ostergaard, E., Sukhatme, G., and Mataric, M. (2001). Emergent bucket brigading. In *Autonomous Agents.*