A Reinforcement Learning Exploration Strategy based on Ant Foraging Mechanisms

Laurens R. Leerink\textsuperscript{a}b, Simon R. Schultz\textsuperscript{b} and Marwan A. Jabri\textsuperscript{b} \\
\textsuperscript{a} Australian Gilt Securities LTD, 37 Pitt Street, Sydney NSW 2000, Australia \\
\textsuperscript{b} Systems Engineering & Design Automation Laboratory \\
Department of Electrical Engineering \\
The University of Sydney \\
NSW 2006, Australia

Abstract

This paper applies recent results from theoretical biology on the topic of ant trail formation and foraging methods to the problem of exploration in a discrete environment with delayed reinforcement. Three mechanisms that have been identified in ant trail formation were implemented as an exploration strategy in the adaptive heuristic critic framework, and applied to a well researched robot navigation task. Simulations indicate that in terms of efficiency the mechanisms used by a single ant Lasius niger perform better than undirected exploration methods, but not as well as specialized directed algorithms. However, when multiple robots simultaneously explore the environment the performance increases in a superlinear manner, resulting in an emergent collective ability larger than that possessed by the individual robots.

1 Introduction

Robot navigation is one of the application areas of learning control where the two opposing aspects of learning are represented in a simply understood manner. On the one hand the robot has to sufficiently explore the environment so that optimal paths can be identified, while at the same time knowledge gained during learning has to be used (i.e. exploitation) if the robot is to avoid making the same mistakes. The trade-off between these two modes of operation has been the subject of some research; a detailed overview is found in [7].

Reinforcement learning (RL) research has gradually developed from solving small maze problems to real-world tasks of controlling robots in limited environments. However, in nature so-called ‘primitive’ insects are continually faced with complex navigational problems in a dynamic environment. Consider a harvester ant carrying a seed in an highly vegetated area, with stems and other obstacles forming physical barriers. Considerable energy and time can be expended if the path chosen back to the nest is sub-optimal. The same RL tradeoffs are found here between exploration (firstly finding food and then the most efficient nest-food path) and exploitation (utilizing the path and avoiding obstacles).

In this paper we attempt to merge knowledge from these two research areas by applying recent results from theoretical biology on the topics of ant foraging behavior to the problem of exploration in a discrete environment with delayed reinforcement. The methods used by the ant Lasius niger in finding the shortest path between nest and foraging area will be discussed, and we will implement these mechanisms as an exploration strategy in a robot navigation task. This task was used by [7] to evaluate a variety of RL exploration algorithms, and as such provides a good comparison between the different methods. Experimental results will then be provided, followed by a discussion and a description of current research.

2 Ants and optimal structures in nature

An overview of optimization problems associated with real or theoretical topological structures is provided by [1] who stated that “any ramified structure found in nature may be considered as a result of such optimization, although in nearly all of them the principle of optimality has yet to be discovered.” The formation of an ant trunk trail (a main trail from which other trails branch) is an example of such an optimal structure. In this structure several aspects of foraging costs (shortest path back to the nest, fighting risks, mobility and the effects of neighboring competitive colonies) are
minimized.

When faced with several paths around an obstacle, ants have no simple means of collective decision making e.g., an hierarchically imposed choice. Instead, *alleldominetic communica* tion is used (my action influences the future actions of others). More information and references regarding this mechanism is provided by [3]. Briefly, this mechanism functions as follows: if an ant comes to a fork in the chemical trail, the probability of it taking a branch is proportional to the amount of trail pheromone on the branch. If the ant then adds pheromone to the trail, the likelihood of that trail being chosen in the future is increased, and very soon the majority of ants will choose that path. This process is analogous to that of 'competitive learning' in neural networks. If one trail is longer than the other, ants taking the longer path will take longer to return, laying down less pheromone per time period at the branch. These small differences are amplified by the autocalylytic or competitive process, with time enabling the selection of the shortest path by the majority of ants.

The ant *Lasius niger* displays a robust capacity for choosing the shortest path between the nest and foraging area, and in addition to the autocalylytic process, three other mechanisms have been identified:

- Bi-directional trail laying. It has been shown by [3] how this mechanism can lead to shortest path selection.
- The use of U-turns: a significant number of ants choosing a branch would turn back before reaching the end, either taking the other branch or returning to the nest [3]. A greater proportion of ants turn back from the longer branch, thereby reinforcing the trail forming process.
- Angle of the path from the main nest-food axis: [4] have shown that the amount of trail pheromone laid down by the forager decreases as this angle increases. This is linked with the fact that several species of ants are known to have strong spatial memories.

In this paper these three mechanisms are used to implement an exploration strategy for RL.

3 Reinforcement Learning and the Robot Environment

In the RL paradigm there is an agent performing actions in an unknown (usually Markovian) environment, receiving rewards or feedback which depend arbitrarily on the past history of actions [2]. The agent usually has access to the system state, and performs the actions at a series of discrete time steps t. The task of the agent is to perform that sequence of tasks that maximizes the total amount of payoff received over the relevant time horizon.

The rule used by the agent to select an action according to the current system state is called the agent’s decision policy. The expected return (discounted payoff) using policy II, starting from the initial state s, and with discount factor γ < 1 is:

\[
V^*(s) = E_{\pi} \left[ \sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s \right]
\]

where \( r_{t+1} \) is the payoff at time \( t+1 \) and \( V^* \) is defined as the evaluation function for policy II and assigns a value to each state s. In the robot navigation task the agent only receives a positive reward when entering the goal state. The evaluation function for state s thus reflects the number of steps required to reach the goal from state s using policy II.

In order to compare the effectiveness of the ant mechanisms as a method of finding the shortest path, the robot navigational task studied by [7] was used. In this task the robot is placed in a maze, and has to be directed from a starting position to a goal. This task has 4485 states which are each represented by a location on the 2-dimensional grid. The robot has 8 actions, each of which moves it to an adjacent location, and receives the following payoffs:

\[
\begin{align*}
    r = \begin{cases} 
        +1 & \text{if the goal state is entered} \\
        -1 & \text{if colliding with a wall (the robot does not move)} \\
        0 & \text{otherwise.}
    \end{cases}
\end{align*}
\]

No a-priori information is given to the robot, and the evaluation values \( V^*(s) \) are initialized to zero for all states s. The problem of finding the shortest path can be seen as obtaining a good estimate of the evaluation function. Once this is available, an optimal policy simply moves from each state to the neighboring state with the highest evaluation value (recall that the evaluation value is directly related to the distance from the goal).

We implemented the abovementioned ant mechanisms in the RL environment as follows: (a) A pheromone value was assigned to each state. (b) The pheromone values decay with time. (c) The amount of pheromone laid down is linearly modulated between zero and one according to the angle between the movement chosen and the main food-nest axis. This information is used from the second trial onwards, when the location of the goal is known.

We implemented U-turns by the following two methods, attempting to model both the intrinsic and trail-based ant U-turns:

- Executing a U-turn when a large change in pheromone level is detected from one state to the other.
Detecting when there is a fork consisting of two or three major branches, and then setting a ‘timer’ corresponding to a certain number of steps. If the timer runs out, the path back to the fork is retraced and the other fork is chosen.

An action evaluation $\text{eval}(a)$ evaluates the 8 possible actions $a$ that are possible in each of the states $s$. We constructed the following action evaluation using $p(s)$, the pheromone value associated with each state $s$ and $f(a)$ the current estimate of the utility value:

$$
\text{eval}(a) = \alpha \cdot f(a) + \zeta \cdot \frac{p(s)}{\max[p | s, a]} + \psi \cdot \text{Random}(a)
$$

with $\alpha = 0.4$, $\zeta = 0.05$ and $\psi = 0.10$ being the linear weighting factors. $\max[p | s, a]$ is the maximum pheromone value of all neighboring states, and is used for normalization. The $\zeta$ parameter was increased linearly to its full value during the first 10 trials to prevent the first trail being repeatedly chosen.

The learning of the utility values was improved by using the ‘experience replay’ method [6]. However, in our implementation the robot was only allowed one step per time period during backtracking.

## 4 Experimental Results

The results will be evaluated according to several measures, the first being the number of actions required for the first trial shown in Table 1. This measures to what extent knowledge from the environment is being utilized; the undirected methods perform the equivalent of a random walk during the first trial.

With one robot the ant based algorithm (ABA) does considerably better than a random search, but not as well as the directed methods. However, if the number of ants is increased, the performance rapidly improves beyond that of all other methods. This is to be expected, as with $N$ ants this is equivalent to taking the best out of $N$ trials. This in itself is useful, but has an even greater effect as learning continues beyond the first trial as discussed later.

The exploration costs and length of the shortest path during learning are shown in Figure 1 for four different numbers of robots in the maze. Exploration costs are measured by the average number of steps per run, and decreases as the best path is more frequently chosen. The shortest path is evaluated using the current estimation of the utility function. Ideally this value should decrease towards the minimal path length of approximately 94. The accuracy of the utility function is measured relative to an absolute benchmark calculated using dynamic programming. As the accuracy improves, the length of the shortest path also decreases.

<table>
<thead>
<tr>
<th>Exploration Technique</th>
<th>Deterministic Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform Distribution</td>
<td>43,000</td>
</tr>
<tr>
<td>Boltzmann distribution</td>
<td></td>
</tr>
<tr>
<td>Semi-uniform distribution</td>
<td></td>
</tr>
<tr>
<td>Counter based exploration</td>
<td>4,700</td>
</tr>
<tr>
<td>Counter/Error based exploration</td>
<td></td>
</tr>
<tr>
<td>Counter-based exploration with decay</td>
<td>5,800</td>
</tr>
<tr>
<td>Recency-based exploration</td>
<td>7,400</td>
</tr>
<tr>
<td>Ant based algorithm (ABA)</td>
<td></td>
</tr>
<tr>
<td>1 Ant</td>
<td>9,486</td>
</tr>
<tr>
<td>5 Ants</td>
<td>3,226</td>
</tr>
<tr>
<td>10 Ants</td>
<td>2,534</td>
</tr>
<tr>
<td>20 Ants</td>
<td>2,197</td>
</tr>
</tbody>
</table>

Table 1: Average number of steps required for 1st run, averaged over 50 trials. Non-ABA values were taken from [7].

## 5 Discussion

With the parameter settings as specified, in all simulations ABA performed in a more efficient manner than undirected algorithms. However, after 20,000 steps the shortest path found at that point in time is strengthened until very little exploration takes place. This behavior can of course be modified by altering the parameter settings. In fact, any of the performance measures shown in Figure 1 can be optimized by appropriate parameter tuning.

Having more ants (or robots) in the maze at the same time results in higher performance according to all three criteria. A larger total number of steps are taken, as each robot takes 1 action per training step. However, the performance increase is more than linear in the number of robots. This indicates that there are constructive interactions taking place, resulting in an emergent collective computational ability that is higher than the sum of the individual robots.

The main reason for this is thought to be the fact that with multiple robots the number of actions required for the first trial decreases (as shown in Table 1) because of the parallel search. The first robot that reaches the goal marks its trail on the way back, so other robots which are still searching can make use of it. This interaction continues, with each subsequent visit to the goal providing better information for the robots currently searching.

The dynamic buildup and decay of pheromones thus provides an environment through which robots can communicate their past actions (failures and successes) to one another, and functions as a common repository...
of information. This is a novel aspect of RL that has yet to be fully explored.

A problem with the above simulations is that there are a multitude of parameters that have to be optimized by hand. Both the learning process and the performance measurement (e.g., determining the shortest path if a state has more than one identical neighboring utility values) are stochastic, implying that several simulations are required to determine the effect of a single parameter change. This is a tedious process, and genetic algorithms (GA) are being considered to automate it. Any of the measurements shown earlier (exploration cost, shortest path or evaluation function accuracy) on their own or in combination can be used to derive a GA fitness measure.

6 Conclusions

Three mechanisms used by the ant Lasius niger in determining the shortest path were implemented as an exploration strategy in a robot navigation task. Simulation results indicate that for a single robot the ant-based strategy performs better than undirected exploration methods, but not as well as specialized directed algorithms. However, when the number of robots is increased a superlinear increase in performance is achieved, indicating a emergent collective ability which in some cases performs better than directed methods.

This research is continuing, with the immediate goal of removing the evaluation function term from the action selection criteria through refinement of the U-turn mechanism. This modification would remove the need to remember the evaluation values associated with each state. We believe that the ability to interact with past actions through chemical modification of the environment is a useful concept, the main advantage being that no memory or an internal representation of the environment is required. This is a novel aspect of RL which has yet to be fully exploited.

Acknowledgments

We thank S. Thrun for making available the software implementing the robot environment.

References


