Reactive food gathering. CLIMA VI contest submission

Robert Logie[1], Jon G. Hall[2], and Kevin G. Waugh[2]

[1] Osaka Gakuin University, Faculty of Computer Science
2-36-1 Minami Kishibe, Suita Shi Osaka, 564-8511 Japan
rob@utc.osaka-gu.ac.jp
[2] The Open University, Department of Computing, Faculty of Mathematics and Computing, Walton Hall, Milton Keynes MK7 6BJ England

Abstract. This short paper describes the design and development of a simple agent system aimed at addressing the food gathering problem set for the 2005 CLIMA contest. Our system is implemented as a collection of reactive agents which dynamically switch between a number of behaviours depending on interaction with their environment. Our agents maintain no internal representation of their environment and operate purely in response to their immediate surroundings. The agents collectively map the environment co-operating indirectly via environmental markers and they use these markers to assist them in locating the depot when they discover food. The required behaviour emerges from the interaction between agents and the marked environment.

The application can be downloaded from:
`http://219.1.164.219/~robert/pwBlog/wp-content/CLIMAbuild.zip`

1 Introduction

Jennings et al. [1] note that a major selling point of purely reactive agent systems is that overall behaviour emerges from interactions between component behaviours and the agent's environment. This inherent simplicity makes reactive agents attractive but it also masks a number of difficulties. The most notable are those of designing agents in such a way that they can take account of non local information and in such a way as to be able to improve their agent level performance over time. Jennings et al. further note that agents using a large number of behaviours can quickly become too complex to understand.

Recent research in normative systems and, particularly, normative reactive systems may provide the means of describing and constraining agent behaviour in a manner which allows us to address this difficulty. For many problems in a tightly bounded environment – problems such as industrial process control or safety systems – reactive agents may be ideal and a fuller understanding of their potential behaviour will be beneficial in allowing their use in increasingly complex scenarios. This contest environment provides such an environment and is, we feel, ideal for the application of reactive agents.

Reactive agents generally operate by having predetermined behaviours or sequences of actions intended to deal with the various circumstances that the agent may encounter. As circumstances change an agent may switch behaviours. Our agent design involved identifying problems within the environment and designing behaviours to address them. This switching between behaviours brings a number of constraints. If an agent's behaviour involves maintaining a record of data and the agent switches to another behaviour, that does not maintain this data, then this data may become outdated. In a dynamic environment such internal world data may be dangerously out of date when the agent returns to using it and maintaining the data my be expensive for a resource bounded agent concentrating on other tasks. We avoid such problems by letting agents use only very local data and data about their internal state or history.

2 The problem — a general approach

We approached the problem by identifying sub-problems which we could associate with agent roles. The roles assigned were those of locating food and, when food has been found, transporting it to the depot. Both of these roles involve searching the environment, the former for food which may be at random locations and the latter the depot which remains in a fixed location. Clearly it is in the system's benefit to have all agents aware of the depot so when one agent finds it some means of indicating its presence to others will be a valuable asset. We have limited communications options by restricting our design to being purely reactive and we limit each agent to being able to carry only one food unit.

We have assumed that the depot location is unknown initially but that it remains in a fixed position throughout a run, when an agent discovers the depot it discovers its permanent location. This leads to a minor difficulty, our agents operate using only very local data and don't know their absolute position in the world [3] which means that they cannot remember an exact depot location. We address this by letting agents leave local markers in the environment. Agents use a random walk to search for the depot. When an agent finds the depot it initialises a "dropper" which allows it to leave a trail of pheromone like weightings on the cells that it visits after having been on the depot cell. Despite the extreme simplicity of this system it allows agents acting only with local data to co-operate in mapping their environment in a way which facilitates the task of carrying food to the depot.

The depot location problem within food transport role is addressed by three agent behaviours; *depot-searching*, the *depot-marking* and the *depot-seeking*. These behaviours, respectively, involve a random walk looking for the depot cell, a random marker laying walk searching for food and a directed pheromone gradient following walk whilst carrying food back to the depot.

[3] Co-ordinate values are only used as a means of keeping agents in bounds and displaying user friendly data. Beyond ensuring that the agent doesn't try to move out of bounds they are not used in any of the agent's operating decisions.

The second search problem – that of finding food – cannot be approached in the same way since food is deposited randomly by the system. Such random placement precludes structured search behaviour by the agents. Food may well appear in a location already searched. The specification indicates that food can only be seen on a cell that an agent is visiting so this rules out giving food a smell that agents can detect. We address this problem with a simple random walk and there is one *food-searcher* behaviour assigned to the food locating role. This behaviour may be concurrent with the depot-searcher and depot-marker behaviours.

Searcher behaviours involve random walks and seeker behaviours involve trail following. There is no food-seeker behaviour so food searching is always a random process and its performance will not improve over time. Seeker behaviours involve using environmental markers left by agents to track previously located objects with persistent locations. It is expected that the performance of seeker type behaviours will improve over time as the environment is more accurately mapped during the random walk of food-searchers.

The food searcher and depot searcher behaviours can operate concurrently. Considering the agent's behaviour in this manner provides a convenient method for analysing behaviour transitions, these are shown in figure 1.

Bonabeau et al. [2, page 26] describe a broadly similar process where ants influence or *recruit* other ants so as to guide them towards persistent food sources, such recruitment based solely on pheromones is known as *mass recruitment*. The depot-marking behaviour is an instance of this mass recruitment as depot-searchers (agents that have yet to find the depot) make use of the pheromone trails from agents that have already located the depot. We have also briefly experimented with other environmental marking methods but felt that these were uncomfortably close to requiring global knowledge or data, something which we are trying to avoid.

3 The agents

Our agents have two modules, a simple reactive core and a move manager. The reactive core senses details of the agent's immediate environment and takes actions depending on its percepts. The agent's roles and component behaviours have been briefly described in section 2. The agent's "cycle" involves sensing its environment, selecting a behaviour, executing that behaviour then making either a directed or random move. Behaviour selection is very dynamic and an agent may switch behaviours on each cycle through its core module. In this environment all behaviours either execute concurrently (such as food-searcher and depot-searcher), or one is suppressed by historic actions (such as having located and picked up food the food-searcher behaviour is suppressed in favour of the depot-searcher behaviour).

Our agent's pheromone tracking behaviour is very simple, finding the depot triggers the agent's depot-marking behaviour causing the agent to prime its trail marker and reinforce any environmental markers in locations it passes through. It

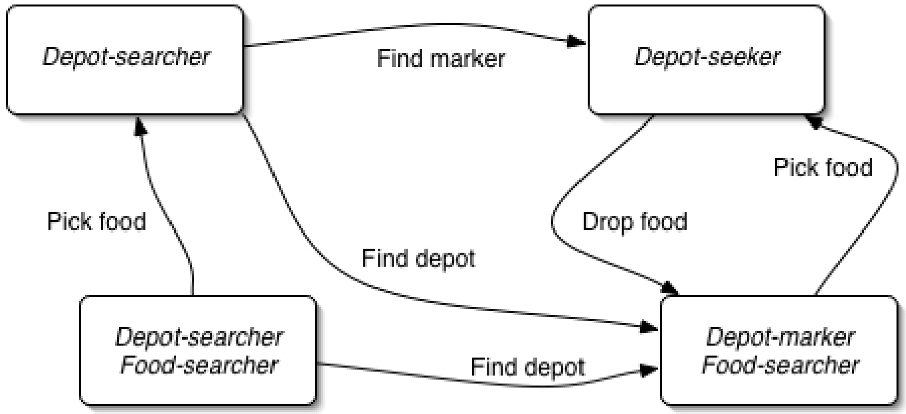

Fig. 1. Agent behaviour transitions.

will reinforce other agent's markers but does not reinforce its own. Finding food will trigger the depot-seeker behaviour causing the agent to stop marking and try to get back to the depot by following marker gradients. The other environment marking methods, mentioned in section 2 that we briefly experimented with were of comparable complexity.

The move manager is coupled to agent core and simply makes sure that the agent doesn't move out of the world's boundaries. This coupling is loose in the sense that the agent doesn't monitor what the move manager does and merely requests a pheromone gradient directed move or a random move. Non determinism caused by the move manager not executing a directed move is handled by the agent's operating in cycles, each cycle is a sense, select, act sequence. This small source of non determinism is probably swamped by the non deterministic aspects of food location in the environment.

4 The problem — logical aspects

Agents *ought* to take food to the depot and they *ought* to do this in as efficient a manner as possible. Considering what agents ought to do allows us to adopt a deontic view of the system but this brings difficulties. Horty notes[3, page 36] that standard deontic logic partitions future worlds into sets of ideal worlds and non ideal worlds. Agents either take food to the depot or they don't, there's no notion of a good or bad way of doing this and, consequently, no notion of improving performance. Norms are typically a social phenomena [4] which makes them intrinsically a multi agent concept but do they have a place in our system? Boella and van der Torre [5] indicate that an important feature of norms is that they allow for behaviour that deviates from ideal and this may allow us to consider norms as a performance improving influence. Our agents are extremely simple, they have fixed transitions between behaviours (see figure reffig:transitions) and no internal systems to allow considered choice, their operation appears to be constrained rather than norm governed. Agents are constrained by their behaviour to drop food on the depot but they may or may not take the best route from where the food was located. Our agents are always capable of taking food back to the depot, a random walk either way would, in this bounded environment a random walk would eventually locate the depot.

When every cell in the world has been marked the agents use a subset of their available behaviours, the depot-searcher is no longer required and transitions are only between depot-seeker and depot-marker/food-searcher behaviours. We think that this can be described as an emergent norm which guides agent behaviour away from the depot-searcher behaviour. If our belief holds and we consider our system as a meta agent then this may be a very simple and possibly degenerate example of of what Boella and van der Torre describe in [6]. Our simple agents delegate the task of improving their performance to an emergent norm and they contribute to its emergence.

5 Observations

Our system performance evaluation had two criteria, the directness of the route taken by agents carrying food back to the depot and whether or not food accumulated in the environment. Test runs were carried out by seeding the environment with a few food units then starting the agents. Initial agent performance is rather poor, agents rely on random searches for both the depot and food. Agents that have found food wander at random and don't appear to be doing anything useful whilst food continues to appear. When one agent finds the depot and begins marking the environment other agents gradually move from random depot-searching to pheromone gradient following depot-seeking. At first this means following, in reverse, another agent's random walk so as to reach the depot. Over a period of time the gradient mapping spreads more widely and agents begin taking more direct routes to the depot with a concomitant performance improvement.

Agents will occasionally becoming trapped by a "livelock". This livelocking manifests itself when an agent appears to walk repeatedly over the same looped path. This only occurs when a food carrying agent is following a pheromone gradient and encounters local maxima. Because the agent follows gradients without backtracking these local maxima may trap the agent. Livelock may be broken by another agent passing through a cell adjacent to the loop and altering pheromone levels sufficiently to allow the trapped agent to escape. The competition system has a small number of agents and if there is a high food density then there is a risk that all four will become livelocked especially where local maxima form within a few grid squares of the depot, a location to which depot-seeker agents are already drawn.

Intuitively if there are non food carrying agents then there is a chance of a livelock being broken. Dealing with this difficulty whilst maintaining a local data only approach is an interesting problem.

6 Installing and running the application

The application can be downloaded from the link in the abstract. This is a zip file which contains seven files, one executable, four DLLs which define agent behaviour, a system configuration file indicating which agents to automatically load on startup and a PDF with brief user instructions. Copy all of these files into a directory on the target machine and run the executable. The program will automatically load the agent DLLs specified in the configuration files and is ready to run.

7 Next steps.

Despite their simplicity our agents do, what we consider, a good job at carrying food to the depot and improving their performance over time. We have concentrated on the depot and not paid much attention to food location simply leaving this to an unstructured, random search. Dealing with the occasional appearance of livelock whilst maintaining a local data only approach presents an interesting problem. Adopting a normative approach we could prohibit livelock. Saying that agents ought not to livelock implies an avoidance capability. One approach is to have "defender agents" [7] which look for possible livelocks and release trapped agents. The difficulty of doing this using only local data is obvious. Our agents are very simple but considering them as a normative system gives a rich view of their interactions and raises a number of questions about how best to improve their performance. If an agent finds food and is unable to pick it up then marking that food location – in a similar manner to the depot – may intuitively seem to be a good step but this may lead to a greater possibility of all agents becoming livelocked.

Our system was developed solely for the CLIMA contest but it has opened up a number of interesting areas to investigate. The observations [1] in section 1 seem to hold even for this very simple system. A normative approach may provide a means of better understanding the interactions in reactive systems.

References

1. Jennings, N., Sycara, K., Wooldridge, M.: A roadmap of agent research and development. Autonomous Agents and Multi-Agent Systems **1** (1998) 7–38
2. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence - from nature to artificial systems. Santa Fe institute studies in the sciences of complexity. OUP (1999)
3. Horty, J.: Agency and deontic logic. OUP (2001)

4. Conte, R., Castelfranchi, C.: Cognitive and social action. UCL press (1995)
5. Boella, G., van der Torre, L.W.N.: Fulfilling or violating obligations in normative multiagent systems. In: IAT, IEEE Computer Society (2004) 483–486
6. Boella, G., van der Torre, L.W.N.: Attributing mental attitudes to normative systems. In: AAMAS, ACM (2003) 942–943
7. Boella, G., van der Torre, L.W.N.: Norm governed multiagent systems: The delegation of control to autonomous agents. In: IAT, IEEE Computer Society (2003) 329–335