Cooperative Behavior, Communication and Task Decomposition in Protective Structure Construction by Neurally Controlled Agents

Frederick L. Crabbe and **Michael G. Dyer**
Artificial Intelligence Laboratory
Computer Science Department
UCLA, Los Angeles, CA 90095
cardo@cs.ucla.edu dyer@cs.ucla.edu

Abstract

This paper describes neurally controlled software agents that construct protective nests as well as other structures out of raw materials gathered from their simulated two dimensional environment. While the agents are inspired from numerous examples in nature, they are not based on any one creature, rather being examples of life as it could be. The agents exhibit several interesting features: collective behavior leads to emergent structures; cooperation between agents leads them to complete otherwise undoable tasks; and communication via the use of limited variable binding creates social task differentiation. This paper describes the neural architecture needed for these features and demonstrates the features in action as the agents build both an intersection of two corridors and a protective briar patch.

Introduction

We are investigating groups of neurally based artificial agents that work together to construct complex objects and protective structures out of raw material found in their simulated 2-D continuous environment. The environment is filled with only circular objects (called discs) and agents which use those discs as building material. In the project, we are concerned with the following issues: the neural controllers necessary to allow the agents to survive and complete their tasks; the nature of distributed algorithms of self-organizing agents that build various nesting components, such as walls; how to direct agents so as to build arbitrary structures, (e.g. for robotic construction tasks); and the designs of agents that can build naturalistic protective structures to aid in their survival. Protective structures are nest-like arrangements of discs that shelter the agents from dangers in their environment, including weather and predators. These arrangements can be enclosures (surrounding walls with an opening), as well other less-structured patterns. Arbitrary structures are any arrangement of discs that can be specified by the researchers, exploring the possibilities of using the agents as real robots to construct useful objects.

This paper will first discuss the cooperative construction of a naturalistic protective structure– a briar patch. The agents arrange heavy discs in a pattern with gaps between the discs in such a way that the gaps are large enough to allow the agents to pass inside, but small enough to prevent the entry of possible larger predators (see figure 1). Because the discs are too heavy for predators to move, the collection of closely spaced discs functions similarly to a briar patch that protects small mammals. The individual discs are also too heavy for a single agent to carry, so pairs of agents must coordinate their behavior so they can work together to move the discs into position. The second part of the paper describes the construction of an intersection of corridors. A foreman agent commands different worker agents to perform different parts of the intersection. The corridors consist of two parallel lines of discs laid by the agents. At the corridor junction, the walls join at right angles to the walls of the other corridor, forming an intersection (figure 2).

Construction in Life

Although the agents have no direct analog in nature, it is common for natural life to exhibit construction behavior (Dyer 1994), and build complex structures (von Frisch 1974). Most ants dig their colonies out of existing material, but other social insects such as bees and wasps synthesize materials to build large multi-roomed nests (Wilson 1971; Hansell 1984). Birds build single and communal nests in a variety of shapes, including bowls, domes and tubes. Social weaver birds build apartment complexes, with up to 100 separate nests, each with a distinct entrance (Morris 1990). In the mammalian world, beavers are the most renowned at building. Their constructions provide shelter as well as modify their environment (Wilson 1968). Perhaps understanding the general issues in simulated agent nest construction will be useful for understanding these natural builders.

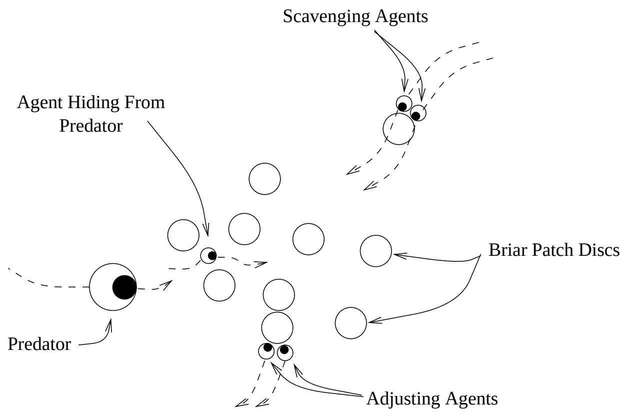

Figure 1: Schematic of a partially completed briar batch. Dashed arrows indicate agent motion. Two agents are bringing in a new disc for the briar patch while another two agents are adjusting the position of a different disc. A fifth agent uses a completed portion of the briar patch to hide from a predator.

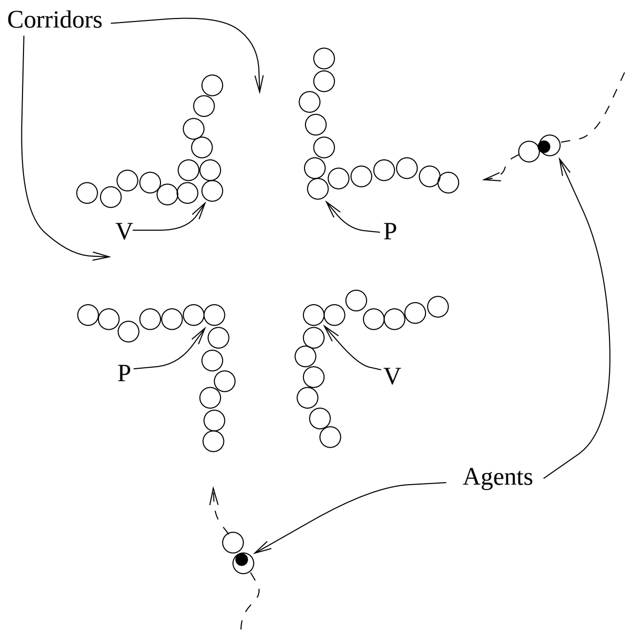

Figure 2: Schematic of a partially completed intersection Dashed arrows indicate agent motion. Two agents shown and six other agents build the eight walls from the center out. Each agent scavenges for one disc at a time to be used in the construction of the wall to which the agent was assigned. The discs labeled P are purple, and the discs labeled V are violet. Use of these colors will be explained in a later section.

Environment

The experiments are run in DiscoTech, a continuous two-dimensional world typically 1000x1000 units in size. Everything in the environment (both objects and agents) are circular discs. Agents are 12 units in diameter. Objects in the environment range from 2 units in diameter to 72 units in diameter. See (Crabbe & Dyer 1998 Submitted) for a more detailed discussion of the DiscoTech environment.

A DiscoTech agent can visually sense other discs in a 120 degree-wide field in front of it. It possesses a set of visual receptor nodes for each color in the environment, e.g. a set for blue (water) discs, a set for red discs and so on. Each set consists of three sensory nodes, one for the left portion of the visual field, one for the right portion, and one for a 4 degree-wide center portion. So with seven colors in the environment, there are 21 visual receptor nodes for each agent. Activation on a particular receptor is $1 - \frac{distanceToObject}{maximumVisualDistance}$.

Along with vision, agents have touch and hearing sensors. The agents can sense if they are touching something and whether they are carrying anything. They also can hear signals emitted from other agents. The agents have an array of signal sensors, with one sensor for each possible type of signal they can hear. When an agent hears a signal, its corresponding sensor gets an activation of 1.

Agent effectors allow agents to: turn left or right with some movement forward (i.e. an agent cannot spin in place); move forward, left, right or backwards; eat; drink; paint discs (i.e. changing the color of a non-food or non-water disc); signal; and lift. Signaling sends a signal to other agents. When an agent signals, the signal is heard by every agent within a certain radius of the signaler. The size of the radius depends on how loud the agent signals. An agent can signal anywhere from a whisper (only agents right next to it receive the signal), to a shout (all agents can hear the signal). Lifting includes grasping a disc. Dropping is done as a cessation of lifting. Agents can only lift one disc at a time; if an agent tries to lift a disc heavier than its strength, the disc is not lifted, but the agent continues to hold on to the disc. If an agent is holding on to a disc that is not lifted, the agent cannot move until it lets go of the disc (stops lifting) or until enough other agents help the first agent lift the disc. Agents can see past (over) discs they are lifting. While more than one agent is lifting a disc, one agent can also slide along the side of the disc while continuing to hold it (figure 3). This is useful when multiple agents are holding a disc and they want to change their relative positions.

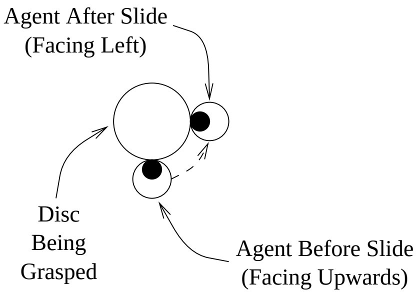

Figure 3: A schematic demonstrating the slide action. The agent is shown facing the disc while holding it. As it slides along the disc (dashed arrow), it changes its relative position to the disc, while still facing and holding the disc.

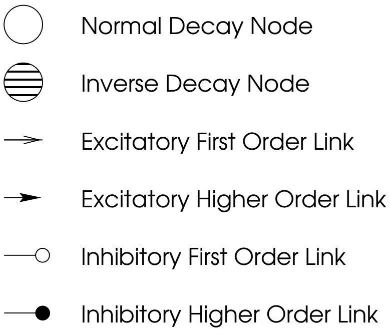

Figure 4: Key for node and link types.

Neural Controllers

Currently the agents are controlled by hand-coded neural networks. In the future we will set the weights and structure of the networks first via adaptive learning, and then via evolution.

Neuron/Connection Types and Dynamics

Agents' neural networks are made up of four types of links and four types of nodes. Figure 4 contains the key for link and node types used in this paper. Nodes spread activation to links which calculate their own activation level before passing it on to other nodes, which sum the activation of all incoming links. Both nodes and links have an activation associated with them, as well as an activation function that is applied to the activation. Having activation functions on links allows us to remove nodes whose sole purpose is to provide an activation function. The equations shown here are used to calculate the activation on nodes and links and are explained below:

$$A_n^{t+1} = f(A_n^t \eta + \beta + \sum_{i=1}^{n} L_i^t), \qquad (1)$$

$$A_n^{t+1} = f((1 - \eta(1 - A_n^t)) + \beta + \sum_{i=1}^{m} L_i^t), \qquad (2)$$

$$L_n^t = f(\beta + A_n^t \omega \prod_{i=1}^{m} H_i^{t-1}). \qquad (3)$$

Nodes Nodes hold some of their activation over time instead of becoming zero before each time-step. Associated with each node is a decay factor between 0 and 1, controlling how much of the activation is held from time-step to time-step. The four node types are: Normal decay, Inverse Decay, Sensor and Effector. They behave as follows:

- All nodes have activation values $0 < A < 1$.
- Normal Decay nodes sum their input from: (a) other nodes, (b) a bias and (c) the node's previous activation multiplied by a decay rate. Then an activation function is applied. Equation 1 is used to calculate the activation of a normal decay node, where A_n^t is the activation of node n at time t, $f(x)$ is the activation function, β is the bias, η is the decay parameter, and L_i^t is the activation of incoming link i at time t. The activation function can be any function, but is typically a sigmoid, step, or linear function.
- Inverse decay nodes behave like regular nodes, except they decay upward towards 1 (see equation 2).
- Sensor nodes take input from the external environment rather than from other nodes via links.
- Effector nodes do not output through links, but rather output to the simulator to designate the set of actions that an agent takes.

Links The links are made up of two classes: *First-Order Links*, and *Higher-Order Links*. In each class, a link can either be excitatory or inhibitory. Higher-order links, like nodes, have activation values $0 < A < 1$, but first-order links can have the value of any floating point number as their activation. First-order links are links whose destination is a node; the activation of the first-order links are summed at a node. Higher-order links modify the strength of either first-order or other higher-order links. Excitatory higher-order links multiply the activation of the higher-order link by the activation of the lower-order link. If the value of an excitatory higher-order link is zero, it effectively turns off the link that it modifies.

Inhibitory higher-order links multiply the activation of the modified link by one minus the activation of the higher-order link. If the activation of the inhibitory higher-order link is one, it turns off the lower-order link; if the activation of the inhibitory higher-order

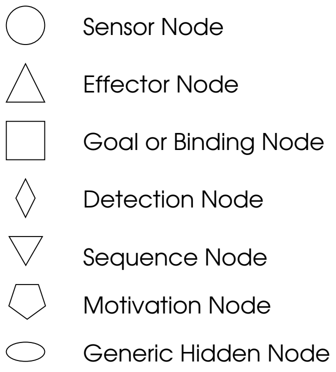

Figure 5: Key for node uses. The shape is a mnemonic device that indicates the use of the node in the network.

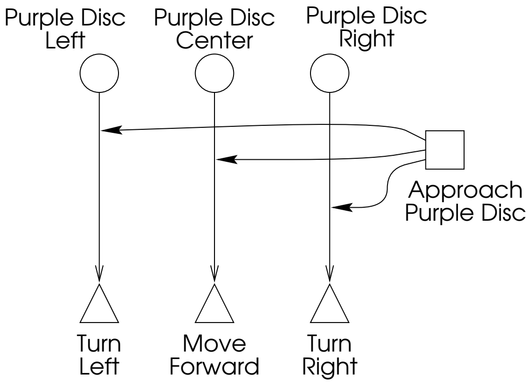

Figure 6: Gated reactive Approach Purple Disc network.

link is zero, then the lower-order link operates as if no modification occurred. Higher-order links act as secondary weights that modulate other links according to activation elsewhere in the network.

All links calculate their activation with equation 3 where L_n^t is the activation of link n at time t then H_i^t is the activation of higher-order link i at time t (if the higher-order link is inhibitory; H is one minus the activation of the link); $f(x)$ is the activation function; β is the bias; A_n^t is the activation of the source node at time t; and ω is the weight. If the link in question is itself a higher-order link, its value will be used to calculate the activation of another link. If the link is a first-order link, then its activation is used to calculate the activation of its target node. If a first-order link has a negative weight, then it is a inhibitory link, otherwise it is an excitatory link.

Basic Level Circuits

In the agents' neural networks, different nodes have different uses. Figure 5 is the key for all of the node types in all of the further diagrams in this paper. Each node type is introduced as it is needed.

Figure 6 shows a reactive network with higher-order connections. Such reactive networks form the basic building blocks of the more complicated networks. Here, the sensory nodes have links directly to the effector nodes, causing the agent to move toward purple discs. When the agent sees a purple disc to the left, the direct connection from Purple Disc Left to Turn Left causes the agent to turn left. When there are multiple purple discs activation might reach both Turn Left and Turn Right, and both actions are taken in parallel. The turns cancel each other out, so the agent ends up moving forward with a slight turn to whichever side has the stronger activation. At the next time-step, one of the purple discs would be slightly closer, so the agent turns even more toward that side, and eventually heads toward the disc in a curved path.

Notice that in figure 6, the links from sensors to effectors are modified by higher-order links from the Approach Purple Disc Goal node. The Approach Purple Disc node represents the agent's goal of approaching a purple disc. When it has activation, it makes the links (from the purple sensor nodes to the effector nodes) become stronger. So if the agent sees a purple disc, the agent is more likely to move toward purple.

In any agent's neural network, there are many of these Goal neural circuits, including: ones to approach each color of object as well as to approach food and water. There are also avoid circuits for avoiding collisions with objects or other agents. Another type of basic circuit is the detection circuit. In these circuits, there is a detection node that becomes active to indicate that a particular external world state is true. These circuits (approach, avoid and detection) are used as subcircuits in other larger circuits to generate more complex behavior. For more on avoidance and detection circuits see (Crabbe & Dyer 1998 Submitted).

Cooperation and Task Decomposition: Building a Briar Patch

In building the briar patch, the agents are divided into two groups. The first group (builder) goes to scavenge a large disc and brings it back to the spot for the briar patch (marked with a brown disc). The first group leaves the disc touching another disc in the briar patch instead of the proper distance away from all the other discs. The second group (adjuster) stays near the briar patch location and when a disc is brought by the first group, the second group adjusts the position of the disc so it is the right distance from the other discs that are already part of the patch, and then they color the new disc brown to make it a permanent part of the patch. Because the briar patch is to offer protection from possible predators, the discs that make up the patch are heavier than a potential predator can move. These heavy discs require two agents to lift and carry it back to the patch. Both the builder and adjusting groups are divided into lead and assistant agents to make the cooperation easier. A lead agent locates a disc and tries to pick it up, and waits until a assistant can come. Then they both pick up the disc and the leader guides the disc to the needed location. Figure 7 shows a lead and assistant builders as they scavenge for discs.

The lead builder first approaches a red disc to scavenge. When it touches a red disc, it tries to pick it up (but fails), colors the disc orange and then waits. The orange color is a sign to the assistant builders that the disc is too heavy and their help is needed. Once a assistant builder has arrived and the two agents can pick up the disc together, the lead builder moves toward the brown disc. As the lead builder pushes the disc, the assistant builder only helps lift it. When the two agents with their disc arrive at a brown disc, the lead builder colors the disc purple, indicating to the adjusting agents that the disc position needs to be adjusted. Then the lead builder begins the sequence again, approaching a new large red disc.

The assistant builder begins by approaching an orange disc (which is being held by a lead builder). When the agent finds and lifts an orange disc, it colors it red so no more agents try to help with that disc (lead builders recognize that the red disc is being lifted and thus do not try to approach it). Then the agent slides counter clockwise around the disc until it is next to the lead builder that is holding the disc. This way both agents are facing the same direction as they carry the disc. As the two agents move the disc, the assistant only helps hold it up, letting the lead builder move it. As a result the agents avoid moving in opposite directions and canceling out each other's movement. The assistant builder continues to lift the disc until it detects that the disc is purple, at which point it lets go and begins the sequence over again.

A lead adjuster's sequence is similar to a lead builder's sequence. First the agent approaches a purple disc. When the agent is next to a purple disc, it lifts it, colors it yellow and waits for an assistant adjuster. Once an assistant arrives and the two agents lift the disc, the lead adjuster moves backwards away from the nearby brown disc. When the lead adjuster detects that it is the right distance away from the brown disc, it drops the yellow disc and colors it brown, making it a permanent part of the briar patch.

Finally, the assistant adjuster behaves like the assistant builder. The agent approaches a yellow disc, lifts it and slides around until it is next to the lead adjuster. It then helps lift the disc until the lead adjuster has pulled it away from the discs it is touching and has colored it brown, at which point the assistant adjuster begins again, looking for yellow discs.

Sequences

In order to perform the complicated sequential behavior described above, the agent must be able to activate and deactivate the goal nodes in the basic circuits in the correct order. For instance, the lead adjuster described above has five steps. Figure 8 shows a sequence circuit for the lead adjuster.

In the Lead Adjust Briar circuit, the *motivation node* (pentagon) feeds activation to the goal nodes (squares) as modulated by the *sequence nodes* (triangles). *Detection nodes* (diamonds) control when one sequence node is turned off and the next is turned on. The process is as follows:

1. The Lead Briar Adjust node begins with some activation, indicating a motivation to adjust the briar discs. None of the sequence nodes have any activation, so the links from the motivation node to the goal nodes do not pass activation.

2. The link from the Lead Briar Adjust node to the first sequence node (S1) has a threshold activation function. If the activation on the Lead Briar Adjust node is over the threshold on the link, S1 becomes active. Sequence nodes do not decay and so remain active until explicitly inhibited.

3. S1 has a higher-order connection to the link from Lead Briar Adjust to Approach Purple Disc, so when the sequence node becomes active, it allows activation to pass from Lead Briar Adjust to Approach Purple Disc. The Approach Purple Disc node controls the agent to move to a purple disc (see figure 6).

4. The link from S1 to the second sequence node (S2) is controlled by the Touching Purple Disc detection node. When the agent is finally touching a purple disc, the detection node becomes active, turning on the link to S2 via a higher-order connection. S2 immediately inhibits S1. It also has an inhibitory higher-order connection to the link from the Lead Briar Adjust node to S1. This prevents the Lead Adjust Briar sequence from restarting before it is completed.

5. The higher-order connection from S2 turns on a link to Lift Disc. Because the goal nodes decay to zero immediately, the Approach Purple Disc node becomes zero, while the Lift Disc node gets activation from

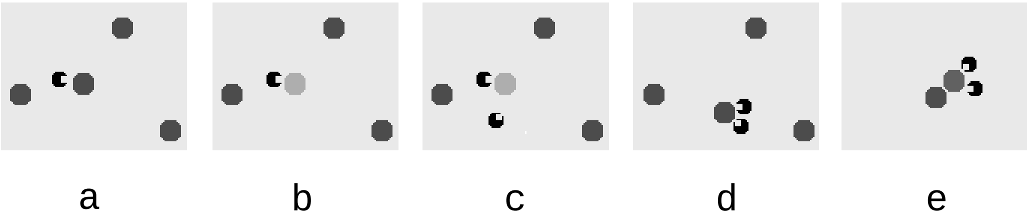

Figure 7: A series of screen dumps of two builders in action. From left to right: (a) the lead builder finds and approaches a red disc; (b) the lead builder colors the red disc orange; (c) an assistant builder arrives; (d) the assistant slides around to be next to the lead as the lead builder moves the disc; and finally, (e) the two builder agents arrive at a brown disc and color the orange disc purple.

Lead Briar Adjust, causing the agent to try to lift the disc.

6. When the agent detects that it is holding a disc (not necessarily successfully lifting), S2 passes activation to S3.

7. The higher-order connections from S3 turn on the links to the Lift Disc and Color Disc Yellow goal nodes. These goal nodes excite the neurocircuitry (not shown) that causes the agent to color the disc yellow while simultaneously continue to try to lift it.

8. When the agent detects that it is holding a yellow disc as a result of the coloring in the last step, the detection node allows the circuit to move to the next step, from S3 to S4.

9. S4's higher-order connections allow activation to pass to Lift Disc and Retreat From Brown Disc. Retreat From Brown Disc causes the agent to back away from any brown discs it sees. If the agent is also lifting a yellow disc (with help from another agent), that disc is carried along with the assistant adjuster.

10. The detection node Disc Held Near Brown Disc which *inhibits* the link from S4 to S5, is active when the disc the agent is holding is near a brown disc. As the agent backs away from the brown disc, the Disc Held Near Brown Disc node turns off, allowing activation to pass to S5, the final step.

11. S5 has a higher-order connection to the link to Color Disc Brown. The agent colors the disc brown, at which point the Touching Brown Disc node becomes active, allowing to link from S5 to inhibit itself. Now, none of the sequence nodes have activation, and the sequence can begin again, as long as the activation of Lead Briar Adjust remains above the threshold on the link from the motivation node to S1.

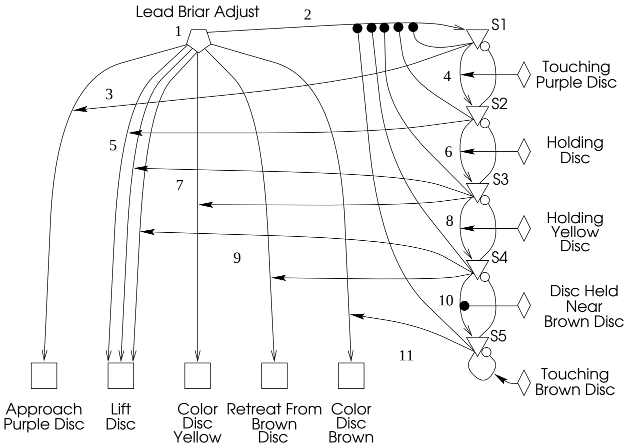

Figure 8: A basic sequence circuit for a lead adjuster. numbers correspond to the steps in §**Sequences**.

Figure 9 shows the sequence circuit for the agent that helps the lead adjuster adjust the positions of the discs in the briar patch.

Results, Issues, and Discussion

Figure 10 shows a briar patch built by two teams made up of the agents described above. While in most runs a similar patch was built, there were occurrences of interesting difficulties. For instance, if a lead builder is holding an orange disc and waiting for an assistant builder to arrive, and at some later time the lead builder becomes hungry enough to abandon what it is doing in order to find food, it then leaves an orange disc behind. Because this disc is now orange, no lead builders will pick it up, but it will attract assistant builders. Because assistant builders can't move the disc on their own, they slide around the disc, never bumping into the lead builder. This sliding will continue until the assistant also gets hungry and leaves, or until the assistant runs into another assistant and, thinking it's a

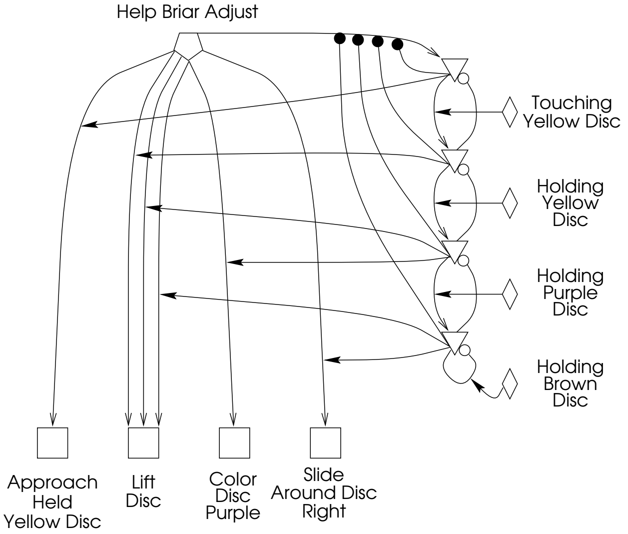

Figure 9: Another basic sequence circuit, this time for a assistant adjuster.

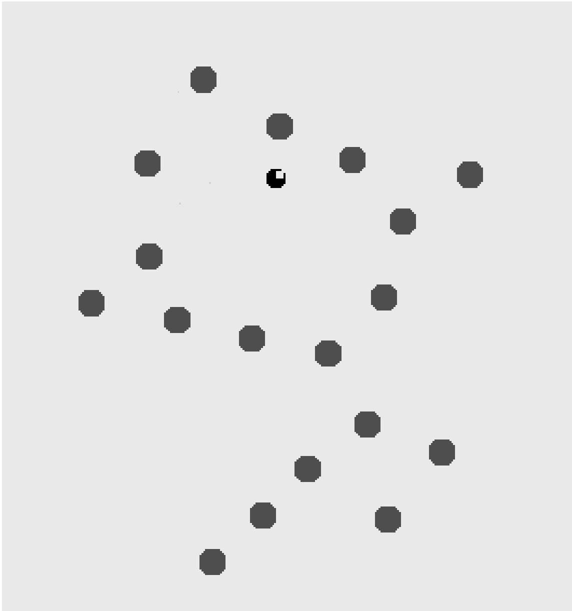

Figure 10: A screen dump of a briar patch built by two teams of two agents each.

lead builder, colors the disc red. In this case, when the assistant builder leaves for food or drink, its disc is back to being red again and some lead builder can go try again on that disc.

The reason the assistant agents only lift, rather than help move the disc as well, is because in several runs, when both the lead and assistant builders were holding the same disc and were moving toward the target disc, they both had a different idea of where the target disc was. As a result they moved at cross purposes and thus the disc moved in neither direction. If instead, the leader were to signal to the assistant which way to go, the assistant could follow the leader and the problem would be averted.

In another common case, when a pair of builders is taking a disc to the briar patch, the motion of the agents and disc causes the assistant builder to collide with another disc in a place the lead builder cannot see, thus trapping the assistant agent and hanging up the progress. A simple "Hey, watch it!" signaled by the assistant builder agent would allow the lead builder to adjust its course for the assistant. These two cases indicate the utility of communication for coordination or cooperation.

Command, Communication, and Binding: Building an Intersection

(Crabbe & Dyer 1998 Submitted) demonstrates agents that build a wall using a colored disc (d1) to mark one end of the wall, and a differently colored disc (d2) to mark the direction in which the wall should be extended. A scavenger agent gathers building material at d2 and colors the material a special color (which alerts a builder to move the material from d2 toward d1). When the builder arrives at d1, the material is placed and colored the same color as d1, making it the new d1 mark. Although the builder has no explicit idea of a wall, because it brought the materials from the area near d2, the wall stretches from d1 towards d2. The building of a corridor intersection uses a similar idea but it is more complex because it involves building eight separate walls, as shown in figure 11.

For building an intersection, the environment contains 9 agents (8 builders and one foreman) and is marked with twelve discs to specify the location of the intersection. The four inner discs mark the junctions of the walls, while the outer eight discs mark the directions in which the corridor walls should grow.

Initially, the foreman goes to each agent and signals which wall that agent should build. It does this by generating a simple signal (which we will mnemonically label "build-wall") followed by a signal for a specific color, meaning that the builder agent should build the wall toward a disc of that color. For each wall, an agent finds a piece of material in the environment and brings it to the disc of the color that it was instructed to by the foreman. From there the agent brings the raw material to the associated inner disc, drops it and colors it the same color as the inner disc. As this is repeated, the agent builds that particular wall. Simul-

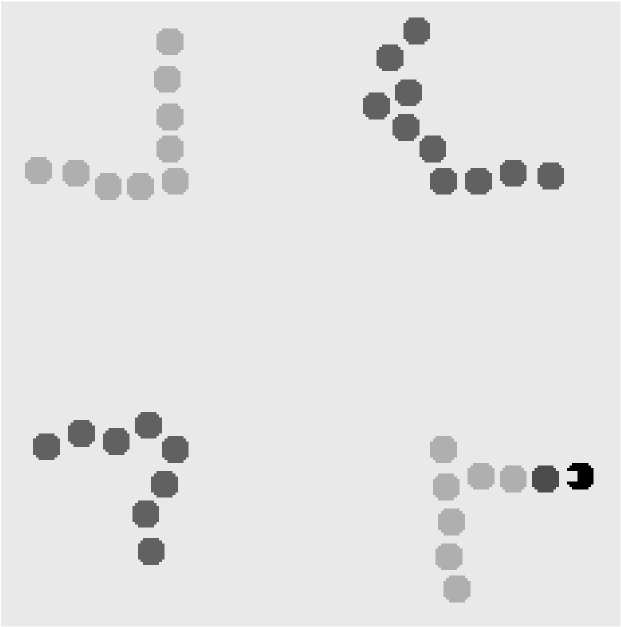

Figure 11: A screen dump of an intersection midway through being built. The intersection is made up of eight walls extending from the four center points to the eight outer discs (not shown).

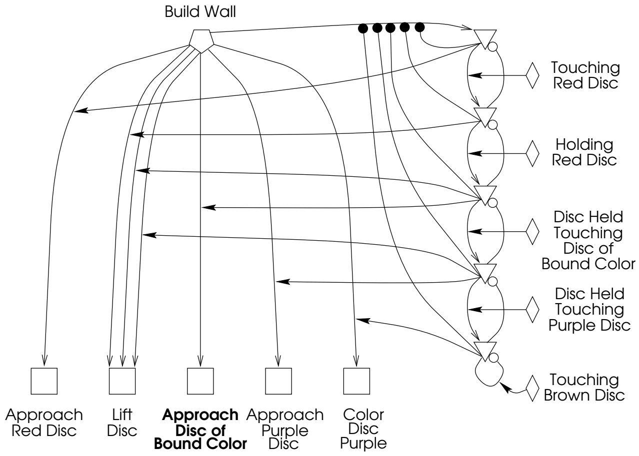

Figure 12: The wall sequence. There are three links from the motivation node to Lift Disc because the agent must hold the disc over three steps, and so there has to be a different link for each step.

taneously, the seven other agents build the seven other walls, thus creating the intersection.

Figure 12 shows the sequence circuit the agents use to build a wall. The agent begins by approaching a red disc. When the agent gets to the red disc, the Touching Red Disc detection node becomes active and the sequence moves to the next step, causing the agent to lift the red disc. As soon as the agent detects that it is lifting the red disc, the sequence moves on to the third step. The agent approaches a disc of the color indicated by the foreman using the Approach Disc of Bound Color goal circuit, while still carrying the red disc. When the agent arrives at that disc, it moves to the next step — approaching a purple disc with the red disc. When it finally is touching a purple disc, the agent drops the red disc it is carrying and colors that disc purple. The sequence then repeats, allowing the agent to build the wall toward the disc of the bound color.

Binding

Up until this point, goal nodes have been tied to a particular color, such as the Approach Red Disc goal node, which, when active, can only control an agent to move toward red discs. This has two disadvantages. First it is inefficient. It requires a large number of goal circuits, and as the number of possible properties of a disc increases, the number of required goal circuits increases exponentially. Second, it makes communication difficult. When the foreman gives orders to worker-agents to build the intersection, it would be desirable to allow the foreman to have one command to tell a worker to build a wall, and eight different commands to say which wall to build. In order to give a step in a sequence the capability to refer to *any* one of the set of colors in the environment, there must be a way to bind that color to a variable and later refer to the value of that variable. This is done with binding circuits.

Figure 13 shows a binding circuit in conjunction with the approach bound circuit. The set of binding nodes (e.g Bind Red and Bind Purple) make up a variable, with one for each possible color value the variable could take. When the Approach Disc of Bound Color node is active, the agent approaches discs of the same color of the bind node that is active. For instance, if Bind Red happens to be active, its higher-order connections allow activation to pass from the red disc sensor nodes to the sensor bind nodes. The Approach Disc Of Bound Color circuit acts just like any other approach circuit, using the sensor bind nodes as input. If instead of Bind Red, Bind Blue were active, the agent would approach blue discs. When both Bind Red and Bind Blue are active, Approach Disc of Bound Color moves the agent toward either blue or red discs. Other basic level circuits also use the sensor bind nodes (such as avoid, and detection circuits).

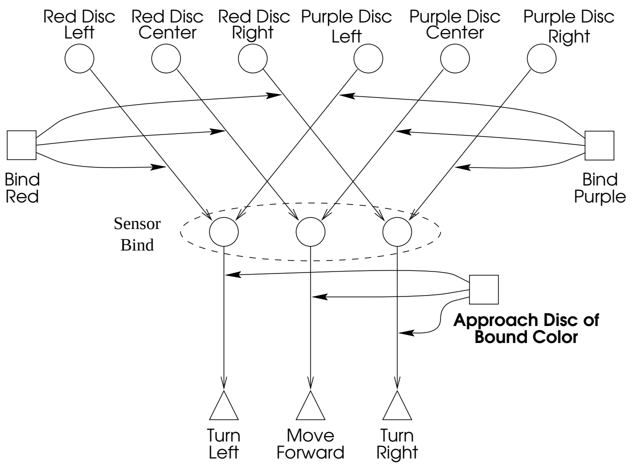

Figure 13: The portion of the binding circuit that binds to red and blue, plus the circuit that the agent uses to approach discs of the bound color. The nodes inside the dashed oval are the Sensor Bind nodes. They act as sensor nodes for the approach goal circuit.

Signaling and Binding

The signals that the foreman gives to the other agents are three-signal commands. First, the foreman gives a command to a worker-agent to attend to what it is signaling. This causes a worker that hears the foreman to pay attention to the rest of the command. Then the foreman signals a command to build the wall, followed by a signal indicating the color that the agent should use to determine the direction in which to extend the wall. It does not matter in what order the foreman gives the two latter signals. Figure 14 shows the circuit that a worker-agent uses to process the foreman's signal. The "Build-Wall" command signal activates a sensor node corresponding to that command. This node sends activation directly to the Build Wall motivation node which controls the sequence circuit for building a wall. The color signal activates an acoustic sensor node corresponding to that signal. The sensor node passes activation on to the binding node for that color, so that when the wall building sequence needs the agent to go to the colored marker, Approach Disc of Bound Color (in figures 12 and 13) will take it to the correct disc. The links from the signal nodes all have a higher-order connection from the Listen goal node. When an agent hears the "attend" command signaled, the Listen node becomes active. This node allows activation to spread from the other signal nodes into the rest of the network. Once Listen is active, anything that the agent hears keeps it active. When the agent stops hearing signals, the activation on Listen decays and the agent will no longer take commands unless it

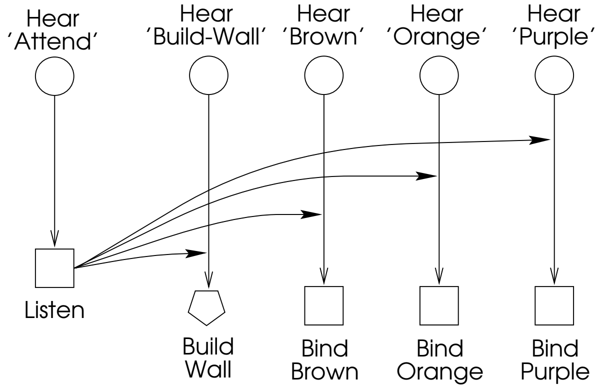

Figure 14: The signal processing sub-circuit of a worker agent. Activation is sent directly from the sensor nodes to the binding and motivation nodes. The links emanating from the motivation and goal nodes have been omitted here.

again hears the "attend" command.

Foreman Agent Sequence

The foreman goes to each agent and tells it which piece of the wall to build. The foreman looks for and approaches an idle agent. The foreman identifies idle agents because they are colored grey (instead of their usual black color). Color is used as an indicator of the agent's state. When the foreman is near an idle agent, it signals to the agent to build the wall and which wall to build. The agent that hears the signal turns itself black as it begins to complete its task. The foreman then looks for another idle agent. This is repeated until all the worker-agents are working (see figure 15).

Issues and Discussion

As shown in figure 2, the center marks are of two different colors (purple and violet) to avoid confusion over which disc to use as the start of a wall. Initially, all four central discs were same color (purple). However, this sometimes caused a problem. When one agent placed its disc first, the agent building the adjacent wall was closer to the first agent's wall than to the mark for its own wall, so it added its disc to the wrong wall (see figure 16). This misplacement only exacerbated the problem, as then that wall was even closer, so the agents ended up building only one wall. To alleviate this problem, the four center discs were made two different colors. This way, agents that are building adjacent walls do not confuse their wall marks.

There is also a danger of confusion when the foreman signals the other agents. If the agents are too close to each other, when the foreman gives a command to one agent, another agent might overhear that command and think that the command is for it. To

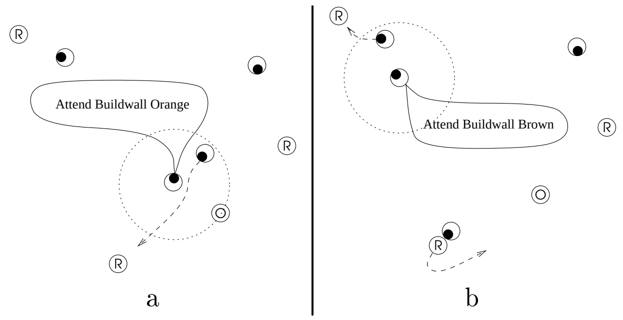

Figure 15: A schematic of the foreman commanding other agents to build an intersection. The dashed arrows indicate a worker-agent's motion after receiving the command. The dotted circle shows the radius in which agents can hear the foreman's signal. (a) a worker's Build-Wall circuit is activated so the worker goes off to get a red disc. (b) The foreman is commanding another worker while the previously commanded worker is bringing red disc toward an orange disc (as the first commend specified).

prevent this, currently the foreman gets very close to each worker before issuing a command. An alternative would be to give agents individual names. When the foreman wants to command an agent it would preface a command with the agent's name instead of the "attend" signal. An agent's signal circuit would look just like the circuit in figure 14, except the Listen node would get activation from a sensor node that is activated when an agent hears its own name. The added complexity of the name-based approach means there is a trade-off between it and the "attend" signal approach. In the name-based approach, a foreman needs to be able to identify and know the name of each agent, requiring more sensory capabilities than currently exists. It also prevents scale-up, because as more agents are added to the environment, the foreman has to recognize more agents individually. If the agents are divided into equivalent groups (i.e consisting of agents with the same name), the foreman can use a naming scheme but does this at the expense of specificity. If the foreman commands an agent of a group, other agents in that group can overhear, resulting in more than one worker working on the task.

After having an agent work on a task for a while, the foreman might want to change what the agent is doing. The foreman give a new command. However, since the old command has already spread activation to many different parts of the network, the agent cannot immediately stop performing the old task. The agent will continue the old task, as well as the new one, until each is completed or until the activation from the commands decays out of the system.

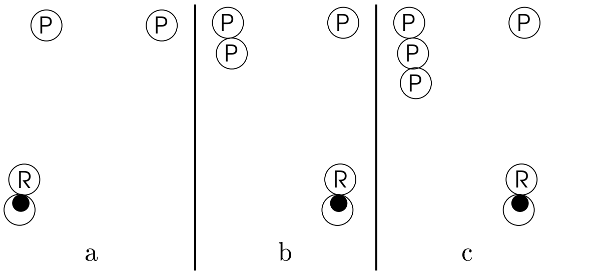

Figure 16: Agents failing to build an intersection. Purple discs are labeled with P, and red discs with R. In (a), the agent assigned to build the left wall brings a disc to the correct purple marker. In (b), the agent assigned to build the right wall begins to approach the right purple marker, but is actually closer to the wall on the left. The agent places its disc on the left wall. In (c), the process continues, with only the left wall getting built.

The interaction between the foreman and the workers is just one example of a broad range of issues in command structure and worker selectivity, where there is a pool of workers from which one or more foremen can recruit workers, and to which workers go when their task is completed. For instance, there could be a hierarchical structure of agents so that high-level agents command lower-level agents which in turn command agents below them in the hierarchy.

The current "language" used by the agents is very simplistic, consisting of an "attend" signal followed by ⟨action⟩ ⟨color⟩ or ⟨color⟩ ⟨action⟩. When agents signal each other, they do not transmit information that is stored in a memory, but rather cause activation to spread in each other's neurocircuits, resulting in changes of behavior.

Future Work

Each sensor, goal, and detection node refers to a specific set of properties of discs in the environment (e.g Red Disc Left, Avoid Purple Disc, and Holding Yellow Disc). Currently, color is the only property that can be visually sensed, keeping the number of nodes small. However, as we increase the number of visually sensed properties, the number of nodes required increases exponentially resulting in a scale-up problem. We intend to address this problem by using distributed vectors on sensors (e.g. using the red-green-blue values to represent color) to reduce the number of sensors. We intend to use more binding circuits to reduce the number of goal and detection nodes.

We plan to add more sensory capabilities so that a foreman could look for an agent that wasn't working (moving, lifting, etc.), instead of relying on the worker-agents changing color. This extra information is also needed for the work we intend to do on learning the signaling language. In order for agents to discern what actions another agent is talking about (e.g. during a language teaching and learning episode), the agents will need to be able to sense the actions of other agents.

Related Work

Therauluz and Banabeau (1995) used collections of simple rule-based wasp-like agents to build nests. Mataric (1997) addresses some of the same issues of cooperation and communication on real robots.

The reactive portions of the circuits described here were inspired by Braitenberg (1984), who proposed them as primitive networks for his vehicles. Werner (1994) showed that adding higher-order connections to Braitenberg networks could improve the behavior of creatures when finding food and water and avoiding predators. Much of our model was inspired by Beer (1990; 1991), who built neural networks by hand to control a simulated cockroach.

Conclusion

The neurally controlled agents described in this paper are able to create a variety of arrangements of discs in their environment – some artificial and others more natural. Through communication (both signaling and by marking materials in the environment) and task differentiation, these arrangements emerged from the collective behavior of the agents.

Acknowledgments

This work is supported in part by an Intel University Research Program grant to the second author. The model is implemented in GNU Common Lisp and runs under Linux on Pentium II based computers, also donated by the Intel Corporation. Software was donated by Microsoft.

References

Beer, R. D., and Chiel, H. J. 1991. The neural basis of behavioral choice in an artifical insect. In *From animals to animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, 247–254.

Beer, R. 1990. *Intelligence as Adaptive Behavior. An Experiment in Computational Neuroethology.* San Diego, MA, USA: Academic Press.

Braitenberg, V. 1984. *Vehicles: Experiments in Synthetic Psychology.* Cambridge, MA, USA: MIT Press.

Crabbe, F. L., and Dyer, M. G. 1998 Submitted. Goal sequence achievement using higher-order neural connection in construction agents. In *From animals to animats5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior.*

Dyer, M. G. 1994. Toward synthesizing artificial neural networks that exhibit cooperative intelligent behavior: Some open issues in artificial life. *Artificial Life* 1:111–134.

Hansell, M. H. 1984. *Animal Architecture and Building Behavior.* London: Longman.

Mataric, M. J. 1997. Using communication to reduce locality in multi-robot learning. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 643–648.

Morris, D. 1990. *Animalwatching.* New York: Crown.

Therauluz, G., and Banabeau, E. 1995. Coordination in distributed building. *Science* 269:686–688.

von Frisch, K. 1974. *Animal Architecture.* New York: Harcourt Brace.

Werner, G. M. 1994. Using second order neural connection for motivation of behavioral choices. In *From Animals to Animats3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, 154–161.

Wilson, L. 1968. *My Beaver Colony.* New York: Doubleday.

Wilson, E. O. 1971. *The Insect Societies.* Cambridge, MA: Belknap/Harvard University Press.