Second-Order Networks for Wall-Building Agents

Frederick L. Crabbe Michael G. Dyer
cardo@cs.ucla.edu dyer@cs.ucla.edu
UCLA Computer Science,
Los Angeles, CA 90095

Abstract

This paper describes robust neurocontrollers for groups of agents that perform construction tasks. They enable agents to balance multiple goals, perform sequences of actions and survive while building walls, corridors, intersections, and briar patches.

1 Purpose

We are developing neurally-controlled agents that can build structures out of building materials found in a two-dimensional continuous environment consisting of circular objects called discs. The agents must complete the sequential building tasks while performing other tasks necessary to their survival, such as eating, drinking or fleeing predators.

In order to perform these construction tasks properly, it is important that our agents' neural control mechanisms exhibit: balance between multiple goals, robust sequencing, backtracking, conflict resolution and opportunism [6]. An agent must be able to perform long sequences of actions. For instance, when scavenging, an agent must look for and find an object to scavenge, go to it, grasp it, look for and find the location to take it to, go there and drop the object.

An agent needs to detect problems in a sequence, including when the agent fails to complete some action or when some other prior state of a sequence becomes undone. When these problems are detected, the agent needs to return to an earlier step in the sequence to correct the problem. For example, if a scavenging agent is carrying a disc back to the scavenge-site when it stops to drink, it must drop the disc it is carrying in order to drink. The agent has to know that it's no longer carrying a disc and must go find another one before returning to the piling site.

An agent should be able to make adjustments in behavior in order to take advantage of opportunities. For instance, if an agent is carrying a disc to a destination and happens to pass near food, it should eat the food encountered along the way.

This paper describes a heterarchical architecture of second-order neural networks that enable agents to perform construction tasks while exhibiting the above properties.

2 Task Algorithms

In order to test the agents' controllers, the agents need specific construction tasks. For these tasks, the agents need algorithms that are executable by the controllers. This section will describe the tasks and the algorithms the agents use to complete them. The agents had three construction tasks to complete: building straight walls; building corridors and intersections; and building briar patches. The agents are handicapped by the lack of a cognitive map. They also lack sophisticated sensors for detecting geometric relations. To compensate, the agents sense the locations of specific colored discs, which are placed by the researchers[1], to locate the key positions for the constructions.

2.1 Wall Construction

A wall is any set of contiguous discs stretching out in a general direction. Corridors are two parallel walls, and intersections are eight walls meeting to form crossing corridors. A briar patch is an arrangement of 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 larger predators.

When building a wall, M agents construct a pile of raw materials by scavenging while N additional agents use the scavenged materials to construct a wall starting from a single disc. The wall is built from the starting point towards the pile of scavenged discs. Figure

[1] We envision that in the future, other agents will set the locations of these initial discs.

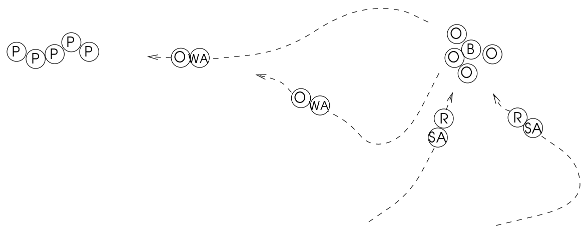

Figure 1: Schematic of the wall-building algorithm in action. While colors are used in the model, here purple discs are marked with Ps, orange discs with Os, and red discs with Rs. Scavenging agents (SA) are bringing red discs to the pile marked with a brown disc (B). Wall-building agents (WA) are taking the the orange discs to the closest purple disc. Agent motion is indicated by the dashed arrows.

1 shows how this works. The scavengers look for red-colored discs to use as building materials. Once a scavenger finds a red disc, it carries it to a single brown disc. The brown disc is used to mark the place where the scavengers are gathering the building materials. Once a scavenger has brought a red disc to the brown disc, it drops the red disc and colors it orange. The orange color signals the wall-building agents that the material is ready for use in construction. If the discs were left red, then the building agents would have to build with red discs, and would not use only the scavenged discs. If the red discs were colored brown like the pile marker, then the building agents would have to build with brown discs and might inadvertently pick-up the brown pile marker and use it to form part of the wall.

When a wall-builder sees an orange disc, it picks it up and looks for a purple disc. A single purple disc is placed in the environment at the start of the wall. Each builder takes an orange disc to a purple disc and then colors the orange one purple. Coloring a disc purple keeps other builders from inadvertently disassembling parts of the wall while building the wall. The coloring of an orange disc to purple as the disc is added to the wall also gives other wall-building agents a new wall end toward which to carry orange discs.

2.2 Briar Patch Construction

In building the briar patch, the agents are divided into two groups. The first group (a pair of builders) goes to scavenge a heavy disc and brings it back to the spot for the briar patch (marked with brown discs). The first group colors the disc purple and leaves it touching another disc in the briar patch instead of the proper distance away from all the other discs. The second group (a pair of adjusters) stays near the briar patch

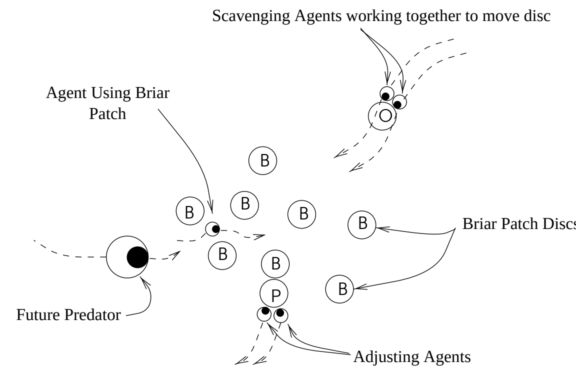

Figure 2: Schematic of a partially completed briar batch. Two agents are bringing in a new disc for the briar patch while another two agents are adjusting the position of a different disc. A fifth agent uses a completed portion of the briar patch to hide from a predator.

and when a disc is brought by the first group, the second group adjusts the position of the purple disc so that 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 briar 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 carry it back to the patch (or to adjust it). 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. When it can't it waits until a assistant can come. Then they both pick up the disc and the leader guides the disc to the needed location. Figure 2 shows builders as they scavenge for discs and adjusters pulling a purple disc back from a brown disc.

The lead builder first approaches a red disc (R) to scavenge. When it touches a red disc, it tries to pick it up (but fails), so it colors the disc orange (O) 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 a brown disc (B). As the lead builder pushes the disc, the assistant builder helps lift it. When the two agents with their disc arrive at a brown disc, the lead builder colors the disc purple (P), 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 (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). 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 (which would cancel 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.

2.3 Intersection Construction via Communication

For building an intersection (figure 3), 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 moves 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. Simultaneously, the seven other agents build the seven other walls, thus creating the intersection.

3 Neural Network Architectures for Construction

Each agent's neural network consists of second-order connections [2], i.e. activation at nodes is calculated by summing the products of pairs of inputs as well as the individual inputs:

$$A_j = f(\sum_{i=1}^{m} w_{ji} x_i + \sum_{i_1=1}^{n} \sum_{i_2=1}^{d} w_{j i_1 i_2} x_{i_1} x_{i_2}).$$

All nodes have activation values $0 \le A \le 1$. Visual and tactile sensors are set by the environment, and actions are performed based of the activations of the motor nodes. There are sets 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. Weights are between $-1 \le w \le 1$.

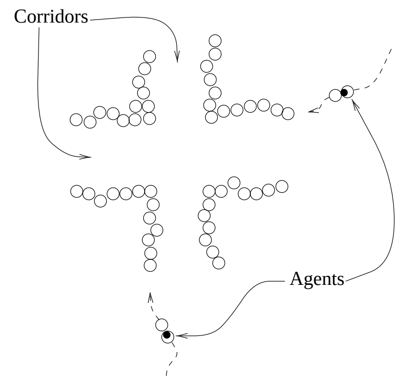

Figure 3: Schematic of a partially completed intersection. Two agents shown (plus six other agents) build the eight walls from the center out. Each agent scavenges for one disc at a time to be used in the construction of the wall to which the agent was assigned by a foreman agent.

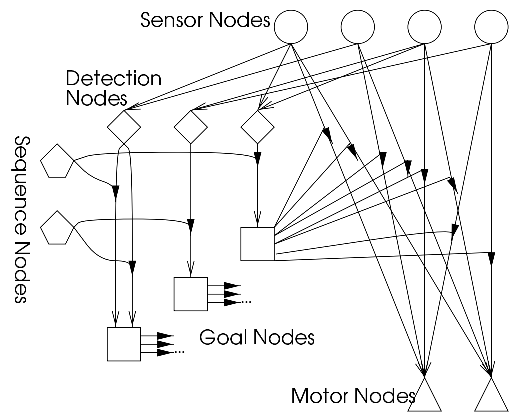

Figure 4: An agent's network is a heterarchy of sub-networks. Second-order connections are indicated by dark arrowheads.

Each network is divided into a layered heterarchy of sub-networks (figure 4). Each sub-network is controlled by a single node which is one of the inputs to all the second-order connections in that sub-network. Thus the controlling node can modulate the amount of effect the sub-network has on agent behavior. Sub-networks higher in the heterarchy control lower sub-networks by changing the activation of the controlling nodes. At the top of the heterarchy are sequence sub-networks, which take input from detection sub-networks, and output into goal sub-networks. The networks are currently

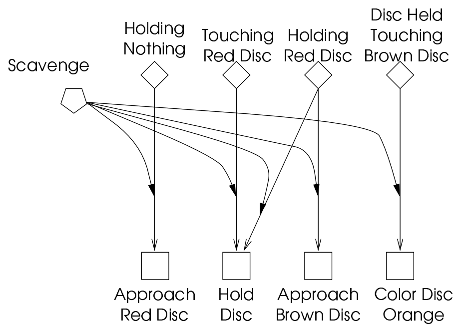

Figure 5: Scavenge sequence network.

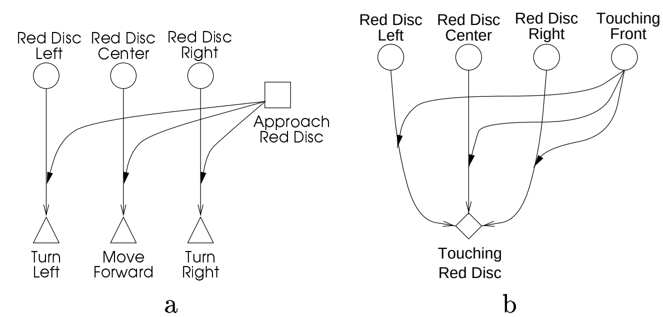

Figure 6: a: Approach Purple Disc network. b: Touching red disc detector sub-network. The higher-order connections from Touching Front ensure that the agent is both near a red disc and touching a disc.

mostly hand constructed, an approach similar to [1].

In order to perform the complicated sequential behaviors described in section 2, the agent must be able to perform actions in the correct order. This is controlled by the sequence sub-networks (figure 5). In sequence sub-networks, the sequence node (here Scavenge shown as a pentagon) modulates the activation passing from detection nodes (diamonds) to goal nodes (squares). Detection nodes, part of detection sub-networks, detect conditions in the environment. Goal nodes control goal sub-networks, which enable the agent to achieve some low-level goal, such as approaching a red disc.

In the sequence shown in figure 5, the agent begins by holding no discs; thus the detection node for that state is active, feeding activation to the Approach Red Disc goal node. The agent approaches and eventually touches a red disc. The Touching Red Disc node becomes active, spreading activation to the Hold Disc node. Now the Touching Red Disc and Holding Nothing nodes lose their activation[2], but Holding Red Disc is active, so the agent holds the disc and approaches a brown disc. When the agent arrives at the brown disc, it colors the disc it is carrying orange. Now it is no longer holding a red disc and so stops trying to hold it. When it drops the disc, it is no longer holding a disc, so Disc Held Touching Brown loses activation and Holding Nothing becomes active, restarting the sequence. Notice that if a goal failure occurs, the state of the detection nodes changes, and the agent automatically moves back to a prior place in the sequence.

The agents have several sequence sub-networks, one for each construction task, as well as ones for survival skills such as finding food. Which sequence nodes are given activation by the researcher at the start of a simulation determines which structures an agent builds.

Figure 6:a shows a goal sub-network with second-order connections. The sensory nodes connect to the effector nodes, causing the agent to move toward red discs when Approach Red Disc is active. When the agent sees a red disc to the left, the connection from Red Disc Left to Turn Left causes the agent to turn left. When there are multiple red 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. Then, one of the red discs will be slightly closer, so the agent will turn even more toward that side, and eventually head toward the disc in a curved path. This resolves conflict at the level of the actions, similar to Braitenberg's Vehicles [3]. The Approach Red Disc node represents the agent's goal of approaching a red disc. When it has activation, it makes the connections (from the red sensor nodes to the effector nodes) become stronger. So if the agent sees a red disc, the agent is more likely to move toward red. Collections of these goal sub-networks control an agent to approach (and avoid) objects in the environment.

In detection sub-networks, detection nodes take input from sensor nodes and become active when a state in the external world is detected. Figure 6:b shows the Touching Red Disc node and its sub-network. Touching Red Disc becomes active only when the agent is touching a red disc. The second-order connections perform conjuncts between the visual and tactile input.

So far, 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 makes communication difficult. When the foreman gives orders to worker-agents to build the intersection, it is desirable to enable the foreman to give one command to tell a worker to build a wall, and eight different commands to say which wall to build. In order

[2] Touching Red Disc is not active when holding a red disc

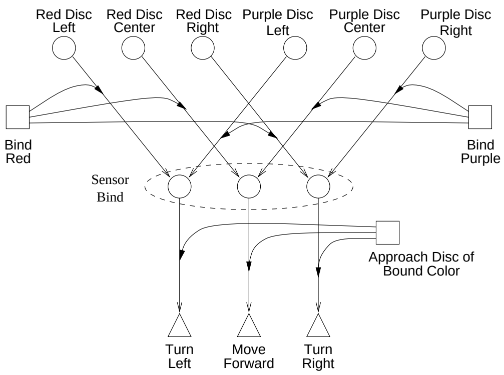

Figure 7: The portion of the binding sub-network that binds to red and blue, plus the sub-network that the agent uses to approach discs of the bound color.

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 currently done with binding sub-networks.

In figure 7 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 can 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 sub-network acts just like any other approach sub-network, using the color 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 sub-networks also use the sensor bind nodes (such as avoidance and detection sub-networks).

4 Simulation Results

We ran experiments for each of the construction tasks, with initial marking discs, raw building materials, food, and water in the environment. During the gathering and building tasks, each agent ate and drank enough to maintain sufficient levels of food and water.

4.1 Wall Construction

We ran the simulation with 5 agents (2 scavengers and 3 wall-builders), 28 red discs to be used as building material, 1 brown disc to mark where the material should be gathered, 1 purple disc to mark the start of the wall, 1 water disc and 16 food discs on a 1080 by 900 world for 30,000 time-steps. In 10 runs a wall emerges each time with a maximum variation (from straight line) within 10 degrees and an average variation within 5 degrees. The foraging agents proceed to collect the red discs and drop them haphazardly around the brown disc (figure 8, right). The other three agents build a wall by extending it from the initial purple disc toward the brown disc (figure 8, left). The wall is uneven because of the variation in the locations of the orange discs. The wall-building agents bring each orange disc straight to the closest purple disc, so deviations in agent starting positions cause deviations in the wall. The spur on the wall in figure 8 was created when an agent became hungry while carrying an orange disc. When the agent got to the food, it dropped the orange disc to eat, and another agent picked it back up again and brought it to the wall. But because the line from the food to the wall was perpendicular to the line from the wall to the brown disc, the agent placed its disc as a spur rather than as a linear addition to the wall.

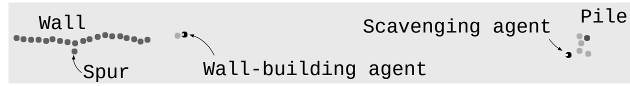

Figure 8: Screen dump of the wall being built. A scavenger is looking for a red disc, while a wall-builder is bringing an orange disc to add to the wall. The other three agents are off the portion of the screen being shown here.

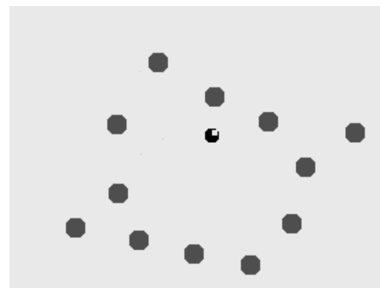

Figure 9: A screen dump of a briar patch.

4.2 Briar Patch Construction

We ran the briar patch simulation with 4 agents (2 builders and 2 adjusters), 28 red discs, and 1 brown disc to mark the start of the patch. In 8 of 10 runs, the agents were able to collect discs into a briar patch suitable for protection, as shown in figure 9. In the 2 failures, discs were arranged at the patch location, but

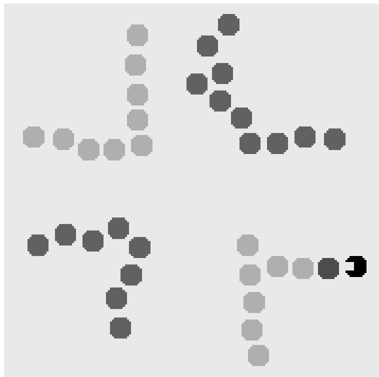

Figure 10: An intersection being built.

there was no area that formed an enclosure.

4.3 Intersection Construction

We ran the intersection simulation with 9 agents (8 builders and 1 foreman), 12 discs to specify the location of the intersection (4 inner discs mark the junctions of the walls, while the outer 8 discs mark the directions in which the walls should grow), and 60 red discs. In 10 runs, the intersection was built (figure 10) each time.

5 Discussion

In the networks described here, *all* the connections were second-order. Use of just second-order connections is derived from the principle that all behavior is both sensor *and* goal oriented. The second-order connections combine both goal and sensor input for the choice of action. For example, in the goal sub-networks, the activation of the goal nodes combines with the activation of the sensors to determine behavior. As goal nodes fluctuate, they dynamically modulate the output [7]. The same is true for the sensor nodes. First-order connections are used to learn the second-order weights via reinforcement as well as vicarious learning, as discussed in [4, 5].

The sequence sub-networks do not explicitly encode a sequence; i.e, they do not keep track of the order of steps explicitly. Instead, they are similar to a set of production rules from world-state to actions. The order of the sequence is preserved by the state of the environment. When an agent is performing a sequence, it knows where it is in the sequence by virtue of which detection nodes are currently active.

The networks described here exhibit a balance between multiple goals. When an agent has multiple goals, several of its goal nodes are active. The choice to pursue a particular goal becomes the product of the goal and the sensor information concerning the satisfaction of the goal. Opportunism is displayed when an agent is close enough to a disc that would satisfy a weak goal. The agent moves to that disc because the strong sensation overwhelms the stronger goals. Each agent performs sequences robustly: it can simultaneously perform multiple sequences, and interrupt one sequence for a more important one, such as the sequence for finding food.

6 Conclusion

The neural architecture described here can control agents to achieve a sequence of goals. The agents reprioritize according to both internal needs and sensory feedback from the environment, taking advantage of chance occurrences in the environment while executing multiple actions in parallel and achieving sub-goals in their proper order. While the behavior of the agents is not optimal (the constructions are squiggly and the agents do not always take the shortest route to a destination), what emerges is naturalistic and robust and is completed by multiple agents working concurrently on distinct subtasks.

Acknowledgements

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

References

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

[2] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, UK, 1995.

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

[4] F. L. Crabbe and M. G. Dyer. MAXSON: Max-based second-order neural network reinforcement learner for mobile agents in continuous environments. Technical Report CSD-900009, UCLA, 1999.

[5] F. L. Crabbe and M. G. Dyer. Vicarious learning in mobile neurally controlled agents: The V-MAXSON architecture. To appear in *Proceedings of the International Conference on Artificial Neural Networks*, Sept., 1999.

[6] T. Tyrrell. *Computational Mechanism for Action Selection.* PhD thesis, University of Edinburgh, 1993.

[7] G. M. Werner. Using second order neural connection for motivation of behavioral choices. In *Proceedings of the third International Conference on the Simulation of Adaptive Behavior*, pages 154–161, 1994.