

Early Evolution of Memory Usage in Digital Organisms

Laura M. Grabowski¹, David M. Bryson², Fred C. Dyer², Charles Ofria², and Robert T. Pennock²

¹University of Texas-Pan American, Edinburg, TX 78539

²Michigan State University, East Lansing, MI 48864
grabowskilm@utpa.edu

Abstract

We investigate the evolution of memory usage in environments where information about past experience is required for optimal decision making. For this study, we use digital organisms, which are self-replicating computer programs that are subject to mutations and natural selection. We place the digital organisms in a range of experimental environments: simple ones where environmental cues indicate that a specific action should be taken (*e.g.*, turn left to find food) as well as slightly more complex ones where cues refer to prior experience (*e.g.*, repeat the action indicated by the previous cue). We demonstrate that flexible behaviors evolve in each of these environments, often leading to clever survival strategies. Additionally, memory usage evolves only when it provides a significant advantage and organisms will often employ surprisingly successful strategies that do not use memory. However, the most powerful strategies we found all made effective use of memory.

Introduction

Organisms must be able to respond to their environment to maximize their chances of survival. They must be able to vary their reactions based on differences in time, place, or circumstance. Evolution has produced many mechanisms that allow such flexible responses, including simple reflexive behavioral routines, such as the response of bacteria like *Escherichia coli* (*E. coli*) to move toward food, or innate behavioral preferences and patterns, as observed in many insects (Dukas and Bernays, 2000). In well-defined, stable circumstances, a repertoire of innate, fixed behaviors may be sufficient to allow organisms to be successful. However, when circumstances can vary due to dependencies on time, place, previous experiences or environmental changes, then more dynamic and flexible behavioral mechanisms are needed. In such cases, memory and learning may allow individuals to more effectively adjust behavior according to the local world state (Dukas, 2008).

How do environment, memory, and learning interact in an evolutionary context? This question is of great interest to both biologists and computer scientists who study the evolution of intelligence. We present early results in our exploration of this interplay in the context of the evolution of

navigation. Our experimental environments are inspired by maze-learning experiments with honey bees (described below). By using these types of environments, we maintained a strong connection between our experiments and their biological motivation, and we were able to probe specific issues relating to the evolution of memory use. Situated at the intersection of biology and computer science, our approach aims to provide insight for both disciplines.

Motivation from insect navigation

Insects are ideal subjects for the study of navigation behaviors. Ants, bees, and other insects use an array of innate strategies to navigate, including *landmark tracking*, where the insect refers to a visual marker (Graham et al., 2003), and *path integration* (Müller and Wehner, 1988), which is the continual internal monitoring of distance and direction relative to a reference location (*e.g.*, the nest). Studies of maze learning in insects are of particular interest, since many bees and ants often follow fixed routes from the nest to a foraging site (Collett et al., 2003). In learning a maze, an insect is learning to follow a well-defined path (Collett et al., 1993). Bees have been trained to fly through mazes of varying complexity. Studies by Collett and colleagues (Collett and Baron, 1995; Collett et al., 1993) used small mazes to investigate bees' ability to learn motor or sensorimotor sequences. One study (Collett et al., 1993) forced bees to fly along prescribed routes and through obstacles in a large box and concluded that bees can remember sensory and motor information that allows them to reproduce a complex route.

A study by Zhang and colleagues (1996) demonstrated that honey bees could use specific visual cues to learn to fly through structurally complex mazes. Another study (Zhang et al., 2000) probed whether bees learn and recognize structural regularity in the mazes. For these experiments, bees were trained and tested in four different types of mazes: constant-turn, where turns are always in the same direction; zig-zag, where each turn alternates direction; irregular, which has no apparent pattern of turns; and variable irregular, where bees had to learn several irregular mazes at the same time. The bees performed best in constant-turn mazes, somewhat poorer in zig-zag mazes, still worse in irregular

mazes, and poorest of all in variable irregular mazes. The authors concluded that the bees' performance in the various configurations depends on the structural regularity of the mazes, and the ease with which the bees can recognize and learn that regularity.

Computational approaches

Evolutionary robotics has dealt extensively with several facets of evolving memory and learning. One aspect is phenotypic plasticity, the ability of a genotype to express differently in different environments. Nolfi et al. (1994) studied this topic by evolving neural network "brains" for virtual robots in environments that alternated between light and dark. Individuals that evolved under these conditions were able to tune their behavior appropriately for both kinds of environments, adapting within an individual "lifetime" to environmental changes.

Evolution and learning employ different mechanisms and occur at differing time scales making their interaction, and, indeed, the evolution of learning, a topic of intense study (Nolfi and Floreano, 2002). A study by Floreano and Urzelai (2000) is a strong example of the latter. They evolved neural networks with local synaptic plasticity and compared them to fixed-weight networks in a two-step task. The networks evolved to turn on a light and then move to a grey square. The results showed that local learning rules helped networks alter functionality quickly, facilitating moving from one task to the other. Blynel and Floreano (2003) explored the ability of continuous time recurrent neural networks (CTRNNs) to solve reinforcement learning problems in the context of T-Maze and double T-Maze navigation tasks, where the robot had to find and "remember" the location of a reward zone. The learning in this case occurred without modification of synapse strengths, coming about instead from internal network dynamics.

Methods

Avida: Overview

Digital evolution (Adami et al., 2000) is a form of evolutionary computation in which a population of self-replicating computer programs, or "digital organisms," is placed in a computational environment where they compete and mutate. Digital evolution can be used both for understanding biological processes and for applying insights from biology to computational problems. The Avida software system (Lenski et al., 2003; Ofria and Wilke, 2004) is a widely used platform for digital evolution. Avida provides a separate instance of real evolution useful for experimental studies (Pennock, 2007).

The "world" in which evolution takes place in Avida is a discrete two dimensional grid containing a population of digital organisms (or "Avidians"), with at most one Avidian per grid cell. The individual organism consists of its "genome," which is a circular list of assembly language-like

instructions, and its virtual CPU. The CPU contains three general purpose registers, several heads, and two stacks. The instructions in the organism's genome execute by acting on the components of the virtual CPU, and execution of instructions incurs a cost in virtual CPU cycles. An Avida organism accomplishes all tasks (*e.g.*, replication and movement) by executing Avida instructions.

An Avida organism replicates by copying its genome into a block of memory that will be its offspring's genome. The copying process is sometimes imperfect, leading to differences between the genomes of parent and offspring. These differences are mutations, and may occur as a substitution, insertion or deletion of an instruction. The Avida instruction set is robust to mutations, so that any program will be syntactically correct even when mutations occur (Ofria et al., 2002). Upon replication, an organism's offspring is placed in a random grid cell, terminating any organism that previously occupied that cell. Thus, organisms in the population compete for the limited space in the set of grid cells, and organisms that replicate more quickly will have a greater number of descendants. An organism can increase its metabolic rate (the relative speed it executes instructions) by performing user-specified tasks. We measure the fitness of an organism as its metabolic rate divided by the number of CPU cycles it requires to replicate.

Experimental environments

Each Avidian was placed in an environment containing a path (inspired by the maze-learning experiments discussed earlier (Zhang et al., 1996, 1999)) that it could gain nutrients by following. To follow a path, an organism must sense cues in the environment that tell it how to stay on the path, and react appropriately to those cues. In some cases, this task necessitated evolving the ability to store and reuse experience. Sensing and movement in the virtual grids were accomplished by executing experiment-specific Avida instructions. The movement instruction moves the organism into the grid cell that it is currently facing. Movement occurs only one step at a time. In the virtual environments of the current study, each organism has its own virtual grid, so organisms do not interact during movement. Orientation changes require additional instructions, one for turning right 45 degrees and another for turning left 45 degrees. Organisms had to combine the different instructions—sensing, movement, and orientation—in order to successfully follow more complex paths.

An organism must navigate its environment to find sparsely distributed "food". Movement requires energy, so each step depletes the organism's energy store. When an organism encounters food, the food gives it more energy than the amount lost through movement. Locations that are off the path are "empty", containing no food. When an organism moves into an empty location, the organism loses a small amount of energy, without regaining any energy.

direction. In this environment, if a turn is in the same direction as the preceding turn, the sense value is different from the sense values of a right turn and a left turn. This new cue signals an organism to “repeat the last turn direction”. This arrangement of information along the path means that an Avidian must be able to change the remembered sense cue value an arbitrary number of times in its lifetime, and at irregular intervals. Thus, this memory is *volatile* as opposed to the unchanging reflex memory needed for the first experimental environment. The arrangement of cues in the second environment type necessitates flexible use of information from an increasingly complex environment. An organism must remember a binary value (turn right or turn left), or one bit of information in information theory terms.

To provide environmental variation and discourage the evolution of brute-force solutions, organisms were presented (at random) with one of four different paths of each environment type during the course of evolution. Thus, any individual organism had a 0.25 probability of being born into the same environment as its parent.

For each experimental environment, we ran 50 replicate populations capped at 3600 organisms for 250,000 updates (a unit of time in Avida), or a median of approximately 33,000 generations. Each experiment seeded the population with an organism capable only of replication. This simple self-replicator ancestor’s genome consists of 100 instructions, comprising a short copy loop and a large number of no-operation instructions. Any other instructions and capabilities can appear through mutations. All experiments used a 0.085 genomic mutation rate for a length-100 organism (a 0.0075 copy-mutation probability per copied instruction, and insertion and deletion mutation probabilities of 0.05 per divide) (Ofria and Wilke, 2004).

Results and discussion

To evaluate the success of different experimental treatments, we used both quantitative performance measures and behavioral tests of evolved organisms. For the quantitative measures of performance, we examined fitness and task quality over time. These values are tracked and recorded during the course of an Avida experiment. For behavioral tests, we traced execution and trajectory of evolved organisms on different path configurations, including paths that were never experienced during the course of evolution.

We use task quality to measure how well an organism performs in a given environment. For this study, task quality measures the fraction of the path an organism traversed, less any movement into empty cells; an organism that traversed the full path without moving into any empty squares would have a task quality of 1.0. Because overall metabolic rate for these experiments was associated solely with the path traversal task, task quality and fitness track closely. The overall performance of a population is shown by the average task quality for that population; the maximum task qual-

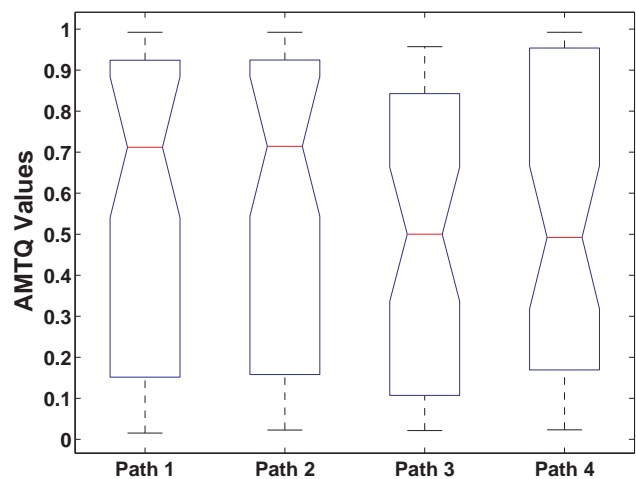


Figure 2: Distribution of average maximum task quality (AMTQ), individual Experiment 1 paths. Paths 1 and 2 are right-turn-only paths, Paths 3 and 4 are left-turn-only paths. There is no significant difference in the AMTQ distributions for each path (Kruskal-Wallis Test, $p = 0.287$).

ity quantifies the performance of the best-performing organisms from each population, and the Average Maximum Task Quality (AMTQ) averages this population maximum task quality over all 50 replicate experiments of each environment type.

To test the behavior of evolved organisms, we ran execution traces for selected final dominant genotypes (most abundant genotype at the end of an evolution experiment) in different environments. With each environment, we tested organisms (1) on the same virtual grids that the organisms experienced during evolution, to observe their behavior in those “ancestral” environments, and (2) in novel environments, *i.e.*, paths that no organism experienced during evolution, to demonstrate the generality of the evolved solutions, or uncover solutions that had been tuned specifically to the ancestral environments.

Evolving reflex actions. Figure 2 shows the distributions of AMTQ values for each of the four single-direction paths. There was no significant difference between the AMTQ distributions for each path, as measured by the AMTQ at the end of evolution (Kruskal-Wallis Test, $p = 0.287$). Figure 3 shows trajectories of the final dominant with the highest ending metabolic rate among all 50 replicate single-direction path experiments, on a right-turn-only path (Figure 3a) and on a left-turn-only path (Figure 3b). The organism’s trajectories on the other two evolutionary environment paths are qualitatively identical to those shown. The organism’s evolved strategy performed well in both turn environments. The organism did some “backtracking” on the right-turn grid, *i.e.*, it turned around and retraced some of its steps on the path. This behavior did not reduce the organism’s task

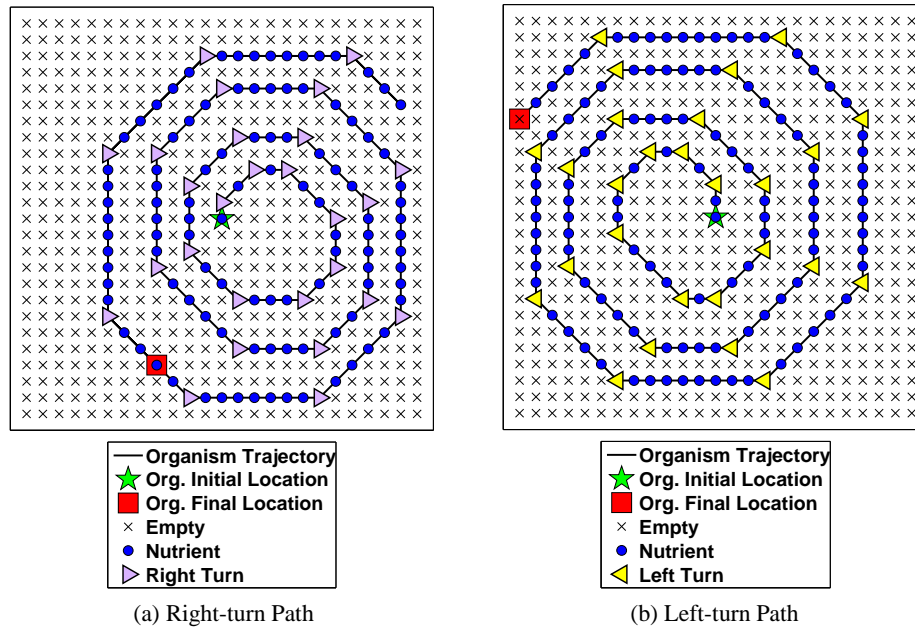


Figure 3: Trajectories of an example evolved organism from Experiment 1 on paths that were experienced during evolution (“ancestral” paths).

quality as the calculation does not penalize an organism for multiple traversals of a path cell. The risk of such behavior is that the organism wastes CPU cycles, thus reducing fitness, although this particular organism still evolved to be the most fit individual in its population. This organism was able to navigate the entire right-turn path without entering any empty cells. The organism also successfully followed the left-turn-only path, stopping after it encountered one empty cell.

To understand this organism’s algorithm, we analyzed its execution while traversing each of these two paths. Most of the path-following and replication code of this organism’s genome is organized into two modules. The first module, “Module 1A,” is mostly concerned with moving on a right-turn path while the second module, “Module 1B,” focuses on left-turn paths and contains a copy loop. These code sections are both executed, regardless of whether the organism is on a right-turn or left-turn path, but the behavior that the modules produce differs according to the path type. In general, Module 1A is a “counting” routine. When the organism is on a right-turn path, Module 1A counts the organism’s steps. On a left-turn path, Module 1A counts the number of rotations the organism executes. Module 1B allows the organism to travel to the end of a left-turn path and then replicate. When the organism is on a right-turn path, the organism uses Module 1B to “backtrack” on the path, retracing some of its steps, while it finishes its replication process.

Evolving volatile memory. The irregular path environment was more challenging than the environments of Exper-

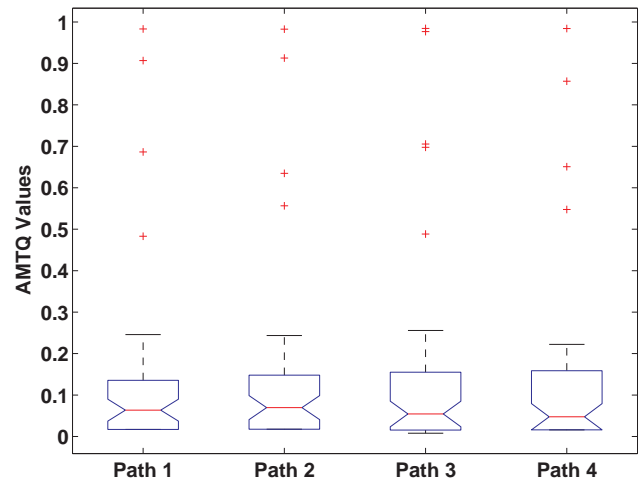
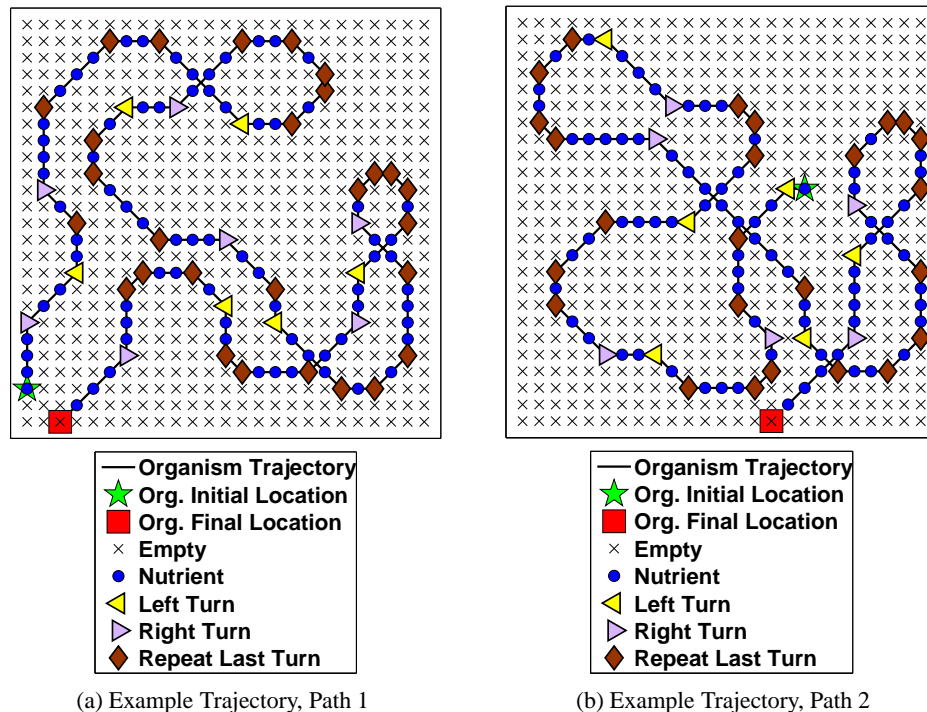


Figure 4: Distribution of average maximum task quality (AMTQ), individual Experiment 2 paths. There is no significant difference in the AMTQ distributions for each path (Kruskal-Wallis Test, $p = 0.238$).

iment 1. The AMTQ for these experiments shows a weaker performance than in the other environment. The difference in AMTQ at the end of 250,000 updates was significantly different in the irregular path experiments compared to the other environment (Kruskal-Wallis Test, $p < 0.05$). There was, however, no significant difference in the performance on each path, measured by the AMTQ at the end of evolution (Kruskal-Wallis Test, $p = 0.238$). Figure 4 shows the



(a) Example Trajectory, Path 1

(b) Example Trajectory, Path 2

Figure 5: Trajectories of an evolved organism from Experiment 2 irregular path experiments. In both (a) and (b), the organism stops moving after encountering one empty cell.

distributions of AMTQ values for each of the four ancestral irregular paths.

Despite the generally inferior performance of the evolved populations in this environment, some highly effective strategies evolved. Figure 5 shows the trajectories of the final dominant organism from the population with the highest AMTQ at the end of the 250,000 update evolution run. This organism has an excellent solution for following these paths, stopping after taking one step off the end of the path into an empty cell. The evolved algorithm is equally effective on novel paths, as shown in Figure 6.

The execution of this organism’s genome is somewhat complicated, and shows an impressive degree of flexibility. In general, this organism operates by moving its execution to different parts of its genome based on the sensed environmental cue. The organism accomplishes all of its path-following with two loops, one for moving through left-turn path sections, “Module 2A,” and the other for moving through right-turn path segments, “Module 2B.” Unlike the other organisms that we have examined in detail, this organism has well-defined functional and structural modularity for handling right-turn and left-turn path sections. Module 2A appears before Module 2B in the organism’s genome. Module 2A can perform an arbitrary number of consecutive left turns, and any number of forward steps. Using Module 2B, the organism can maneuver through right-turn path sections. Module 2B functions with arbitrary numbers of

forward steps and repeated right turns. If a left turn cue is sensed, Module 2B terminates and execution jumps to the beginning of the genome, eventually reaching Module 2A again. If an empty cell is sensed while execution is in Module 2B, the module terminates and execution continues with the instructions after the module. In addition to the movement modules, the organism has a tight copy loop near the end of its genome that accomplishes almost all the copying for the organism’s replication.

There are two features of this organism that are particularly interesting. The first is the organization of the genome. The sections of the genome that do the bulk of the work for this organism—the two movement modules and the copy loop—are functionally and spatially modular. For all three of these loops, very little happens within them apart from the main function of the loop. The loops are also spatially modular: they are located in different sections of the genome. Example organisms from the preceding experiments also demonstrate structural modularity, but their functional modularity is generally less defined. The second feature of special interest is the flexibility of execution flow between code modules. The execution flow enables the organism to cleverly handle all the contingencies of the environment. For example, even though Module 2A (left-turn module) is encountered first in the sequential execution of the genome, if a right turn is encountered first, the flow moves easily through Module 2A and into Module 2B (right-turn module). The

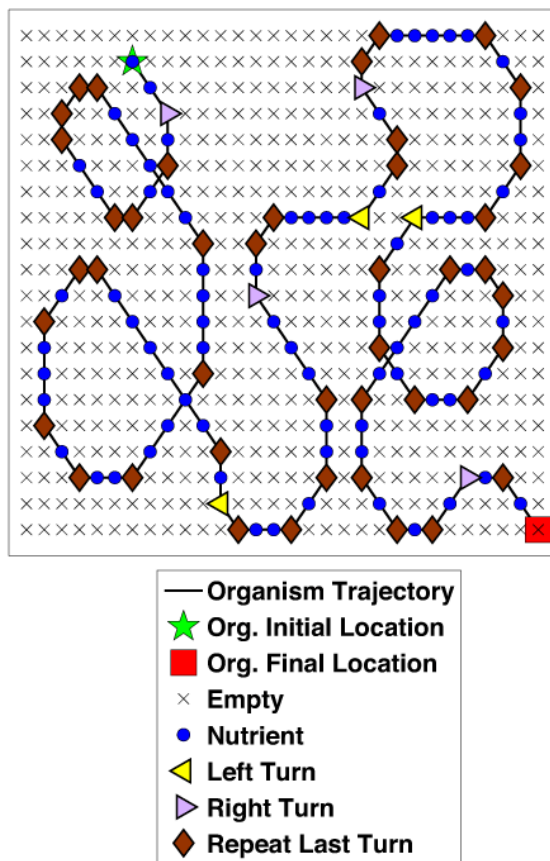


Figure 6: Trajectory of an evolved organism from Experiment 2 irregular path experiments, traversing a novel path.

algorithm evolved to deftly maneuver along the paths, using environmental cue information to alter its execution.

By analyzing the execution of evolved genomes from both environment types, we found that memory use involved both the organization of the genome and volatile states of the organisms' virtual CPUs. The organization of the genomes provided functional modularity, while different environmental information created different states of the virtual CPU that lead to differential behavior based on the current state in the environment. The resulting behaviors formed a simple set of behavioral repertoires that could be used flexibly in response to environmental stimuli.

Conclusions and Future Work

Through these results, we illustrate that memory and flexible behavior can evolve in simple environments. Evolution capitalizes on both environmental change and regularity to construct these solutions. The experiments presented here suggest, not surprisingly, that it is more difficult to evolve volatile memory than to maintain "evolutionary memory" (reflexes).

Results such as those we present here may inform inves-

tigation in both biology and computer science. Insights into the evolution of behavioral characteristics of natural organisms must rely on studies of extant species, since the fossil record provides little information about an animal's behavior. Our results may help provide additional insights by allowing detailed analysis of the evolutionary transitions that led to intelligent behavior. Those insights can, in turn, be used in the context of computer science to produce artificial systems that exhibit the behavioral flexibility of natural systems. The current work is an early step in this direction.

Natural evolution produced many impressive navigation abilities in animals. These capabilities are made up of interwoven strategies, which are themselves made up of simpler underlying mechanisms. Memory is undoubtedly one such underlying mechanism. We witnessed memory evolve even when not required in the single-direction path experiments; the "step-counter" organism based part of its strategy on tracking its progress along its path. This organism possesses a simple odometry mechanism, like those found in many animal navigation systems. This same organism was also able to count its rotations to orient itself in the correct direction. Self-referential compasses are another component of animal navigation. The results from our study hold promise of future insights into questions surrounding the evolution of navigation. For example, the environments used in the current study can be adjusted so that organisms need to explore the environment to find resources, and then return to their initial location as efficiently as possible. This situation sets up investigating the evolution of path integration. There is a rich collection of evidence of this ability in many animals, and different models of the mechanism have been presented (e.g., Mittelstaedt (1985), Müller and Wehner (1988), Hartmann and Wehner (1995)). How evolution produced such a capability is, however, an open question. Some interesting work has explored this issue, such as Vickerstaff and DiPaolo (2005), who used a genetic algorithm approach to evolve neural network models of path integration. Experiments such as those in the current work have the potential to contribute to that discussion, by allowing detailed examination of both the evolution and the evolved algorithms that are not possible in network based approaches.

The path-following environments can be used to study the evolution of associative memory, the process by which animals learn about cause-and-effect relationships between events and then behave appropriately (Rescorla, 1988; Shettleworth, 1998). We can simulate the arbitrary stimulus, important for associative learning, by generating random numbers for signpost cues each time a particular path is assigned to an organism, changing the values for the organism's offspring. For true associative memory, the organisms should be able to associate arbitrary features of their surroundings with their desired goal. We plan to vary the relationship between the cue and the target, so the cue might be prompting a turn in the paths, or it might indicate that the food source is a

certain distance ahead, regardless of what else the organisms have seen in the interim.

The experimental results that we present here demonstrate the evolutionary origin of simple intelligence and behavioral flexibility. Organisms from these experiments were capable of gathering information from the environment, storing that information, and using the information for decisions. Moreover, organisms that succeeded in the irregular path environments were able to use a past individual life experience to guide future decision-making.

Acknowledgments

We would like to thank Philip McKinley, Wesley Elsberry, Jeff Clune, Michael Vo, Erica Rettig, and other members of the MSU Digital Evolution Laboratory for valuable comments and discussion. This work was supported by a grant from the DARPA FunBio program, NSF grant CCF-0643952, and a grant from the Cambridge Templeton Consortium, "Emerging Intelligence: Contingency, Convergence and Constraints in the Evolution of Intelligent Behavior."

References

- Adami, C., Ofria, C. A., and Collier, T. C. (2000). Evolution of biological complexity. *Proceedings of the National Academy of Science*, 97:4463–4468.
- Blynel, J. and Floreano, D. (2003). Exploring the T-maze: evolving learning-like robot behaviors using CTRNNs. In Cagnoni, S., editor, *Applications of Evolutionary Computing*, Lecture Notes in Computer Science, pages 593–604. Springer, Berlin.
- Collett, T. S. and Baron, J. (1995). Learnt sensori-motor mappings in honeybees: interpolation and its possible relevance to navigation. *Journal of Comparative Physiology A*, 177:287–298.
- Collett, T. S., Fry, S. N., and Wehner, R. (1993). Sequence learning by honeybees. *Journal of Comparative Physiology A*, 172:693–706.
- Collett, T. S., Graham, P., and Durier, V. (2003). Route learning by insects. *Current Opinion in Neurobiology*, 13:718–725.
- Dukas, R. (2008). Evolutionary biology of insect learning. *Annual Review: Entomology*, 53:145–160.
- Dukas, R. and Bernays, E. A. (2000). Learning improves growth rate in grasshoppers. *Proceedings of the National Academy of Science*, 97(6):2637–2640.
- Floreano, D. and Urzelai, J. (2000). Evolutionary robots with online self-organization and behavioral fitness. *Neural Networks*, 13:431–443.
- Graham, P., Fauria, K., and Collett, T. S. (2003). The influence of beacon-aiming on the routes of wood ants. *Journal of Experimental Biology*, 206:535–541.
- Hartmann, G. and Wehner, R. (1995). The ant's path integration system: a neural architecture. *Biological Cybernetics*, 73:483–497.
- Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423:139–144.
- Mittelstaedt, H. (1985). Analytical cybernetics of spider navigation. In Barth, F. G., editor, *Neurobiology of arachnids*, pages 298–316. Springer, Berlin.
- Müller, M. and Wehner, R. (1988). Path integration in desert ants, *Cataglyphis fortis*. *Proceedings of the National Academy of Sciences of the United States of America*, 85:5287–5290.
- Nolfi, S. and Floreano, D. (2002). Synthesis of autonomous robots through evolution. *Trends in Cognitive Sciences*, 6(1):31–37.
- Nolfi, S., Miglino, O., and Parisi, D. (1994). Phenotypic plasticity in evolving neural networks. In *Proceedings of the PerAc '94 Conference*, pages 146–157, Los Alamitos, CA. IEEE, IEEE Computer Society Press.
- Ofria, C., Adami, C., and Collier, T. C. (2002). Design of evolvable computer languages. *IEEE Transactions in Evolutionary Computation*, 17:528–532.
- Ofria, C. and Wilke, C. O. (2004). Avida: a software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229.
- Pennock, R. T. (2007). Models, simulations, instantiations, and evidence: the case of digital evolution. *Journal of Experimental and Theoretical Artificial Intelligence*, 19(1):29–42.
- Rescorla, R. A. (1988). Behavioral studies of Pavlovian conditioning. *Annual Review of Neuroscience*, 11:329–352.
- Shettleworth, S. J. (1998). *Cognition, evolution, and behavior*. Oxford University Press, New York.
- Todd, P. M. and Miller, G. F. (1990). Exploring adaptive agency II: simulating the evolution of associative learning. In *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior on*, pages 306–315. MIT Press, Cambridge, MA, USA.
- Vickerstaff, R. and Di Paolo, E. A. (2005). Evolving neural models of path integration. *Journal of Experimental Biology*, 208:3349–3366.
- Zhang, S. W., Bartsch, K., and Srinivasan, M. V. (1996). Maze learning by honeybees. *Neurobiology of Learning and Memory*, 66:267–282.
- Zhang, S. W., Lehrer, M., and Srinivasan, M. V. (1999). Honeybee memory: navigation by associative grouping and recall of visual stimuli. *Neurobiology of Learning and Memory*, 72:180–201.
- Zhang, S. W., Mizutani, A., and Srinivasan, M. V. (2000). Maze navigation by honeybees: learning path regularity. *Learning and Memory*, 7:363–374.