

On the Evolution of Motility and Intelligent Tactic Response

Laura M. Grabowski
Dept. of Computer Science
and Engineering
Michigan State University
East Lansing, MI, 48864, USA
grabow38@msu.edu

Wesley R. Elsberry
Lyman Briggs College
Michigan State University
East Lansing, MI, 48864, USA
elsberry@msu.edu

Charles Ofria
Dept. of Computer Science
and Engineering
Michigan State University
East Lansing, MI, 48864, USA
ofria@msu.edu

Robert T. Pennock
Lyman Briggs College and
Depts. of Philosophy and
Computer Science and
Engineering
Michigan State University
East Lansing, MI, 48864, USA
pennock5@msu.edu

ABSTRACT

We present our first results concerning the *de novo* evolution of motility and tactic response in systems of digital organisms. Our model organism was *E. coli* and the behavior of interest was gradient following, since this represents simple decision-making. Our first experiments demonstrated the evolution of a tactic response, both when provided with a hand-coded system to remember previous gradient concentrations and without this crutch where the organisms must determine how to store previous values on their own. In our second set of experiments we investigated two different rotation strategies, random and systematic, and found no significant performance difference between the two strategies. These experiments served as a stepping-stone and proof-of-concept of the infrastructure needed for our future work on the evolution of simple intelligence.

Categories and Subject Descriptors

I.2.0 [Computing Methodologies Artificial Intelligence]: General; I.6.0 [Computing Methodologies Simulation and Modeling]: General

General Terms

Experimentation

Keywords

Digital evolution, chemotaxis, Avida, gradient following, experimental evolution

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '08, July 12–16, 2008, Atlanta, Georgia, USA.
Copyright 2008 ACM 978-1-60558-131-6/08/07 ...\$5.00.

1. INTRODUCTION

1.1 Intelligence and evolution

Computer scientists have long worked toward the goal of producing computational systems that are capable of flexible, complex behaviors. This goal has proven elusive, since the design of such intelligent systems has been remarkably difficult. Historically, the fields of philosophy and artificial intelligence (AI) have both addressed the question of intelligence with a focus on what may be termed propositional intelligence (knowing that) rather than behavioral intelligence (knowing how). This attention to high-level, human-like intelligence and cognition has led many AI researchers to use top-down methods to produce hand-designed, special-purpose solutions to particular problems. Despite some impressive successes, the promise of AI to produce general-purpose, flexible systems remains largely unfulfilled.

An evolutionary standpoint can provide a different perspective on the problem of intelligence. Evidence indicates that, just as the modern human body evolved from other, earlier morphologies, human intelligence evolved from other pre-existing forms of intelligence. Therefore, instead of undertaking the daunting task of designing an artificially intelligent system from the top down, perhaps a more advantageous approach is to create the circumstances in which intelligence can evolve, and allow the process of evolution to find the design.

The field of AI has explored various types of cognition, often taking inspiration from animals, both vertebrates and invertebrates. However, even relatively simple brains, such as those of insects or nematode worms, may have complex interconnections that, although deeply studied, remain incompletely understood. The idea at the center of the current study is to step far back in evolutionary history, and far down in complexity, to examine behaviors and mechanisms that demonstrate intelligence in a much simpler form: simple decision-making. Briggman *et al.* [10] propose that decision-making can apply to a “spectrum of goal-driven behaviors,” ranging from complex (conscious choices with expectation of reward) to simple (predictable reflexes). A motile organ-

ism's ability to orient relative to sensory stimuli is among such reflexive behaviors. A fundamental form of orientation is following gradients—the change of some property over distance—in the environment. The ability to navigate environmental gradients is essential for even the simplest motile organisms, and is likely a building block for more complex movement-related behavior, such as navigation.

Here, we report our initial steps in exploring the *de novo* evolution of motility and tactic response in systems of digital organisms, and a group of first experiments that provide proof of concept and point toward future work. This work is only the first step in a much larger research initiative, and is centered on building infrastructure that will facilitate the ongoing investigation of simple intelligence. We focus on gradient following as simple decision-making, using *E. coli* as the model organism. There are a number of compelling reasons for using an organism like *E. coli* as the starting point for investigating decision-making. It is a simple, well studied, and highly successful organism. More importantly, in relation to decision-making, *E. coli* is able to obtain information from its environment and use the information it gathers in order to survive, moving through the environment and orienting to sensory stimuli.

1.2 Tactic responses and bacterial chemotaxis

In biology, gradient following is often referred to as *tactic response* or *taxis* (plural *taxes*) [11]. Motile microorganisms and cells show sensory responses to a range of environmental stimuli, including light, sound, gravity, magnetic fields, and chemical signals. The distinction is often made between taxes, applied to motile cells and organisms, and tropisms (an in-place orientation toward or away from a stimulus), applied to organisms that are firmly attached to a substrate and so are incapable of locomotion. Taxis may be either positive, in which movement is toward the stimulus, or negative, involving movement away from the stimulus. Taxes are also often classified on the basis of the stimulus, such as light (*phototaxis*), heat (*thermotaxis*), or chemical agents (*chemotaxis*) [11]. In a broad sense, taxes function to move the cell or organism toward substances or conditions that are in some way beneficial, and away from situations or agents that may be harmful.

Chemotaxis in bacteria was first described by T.W. Engelmann and W.F. Pfeffer [3, 7] in the late 19th Century, but its molecular mechanisms were not understood until the 1960's [2, 8]. Chemotaxis may be an ancient process, predating the evolutionary divergence of eubacteria (gram-negative, gram-positive, blue-green algae) and archebacteria (e.g., methane bacteria) [25]. Over a century of research has provided many important insights into the details of bacterial chemotaxis.

The chemotactic response is driven by attractants, substances that the bacteria tend to move toward, and repellants, substances that the bacteria tend to move away from. When no attractant or repellent is present, or when the concentration of attractant or repellent is uniform—that is, when there is no gradient—a bacterium such as *E. coli* tends to swim in a relatively smooth, straight line for a number of seconds (called a “run”); it then “tumbles” for a fraction of a second, rolling and spinning, and then swims in a straight line again, now in a new, random orientation. In contrast to this behavior, bacteria tumble less frequently when encountering increasing concentrations of an attractant (i.e., they swim longer runs), and tumble more often when the

attractant concentration is decreasing [4, 8]. Simply put, bacteria tend to swim in the same direction for longer periods of time when the situation is improving, but will return to the “random walk” behavior if the situation is not improving [7]. This behavior applies both in a spatial gradient (i.e., a higher chemical concentration on one side than on the other) and in a temporal gradient (i.e., a higher concentration now than earlier). Evidence indicates that, for small cells such as *E. coli*, the sensing itself is temporal, not spatial; that is, concentrations are sensed over time, instead of comparing two simultaneous sense values at the front and back of the cell [7]. Some researchers regard this as akin to a simple “memory:” the cells can compare the present with the past, and respond accordingly [13].

1.3 Taxis inspired methods in computation

Tactic response has served as the impetus for many computational approaches that vary widely in their aims, inspirations, and methods. The pervasiveness of tactic responses in animals motivates biomimetic approaches to robot control and navigation. Such biomimetic approaches focus either on capturing aspects of the animal's behavior for some engineering application, or on testing hypotheses about mechanisms related to the animal's behavior. Chemotaxis in lobsters has been implemented on robots, both for control that emulates the lobster's performance for underwater tasks applied to such functions as autonomous mine countermeasures [5], and to test hypotheses about the lobster's chemotaxis algorithms [18]. Morse *et al.* [21] implemented an autonomous robot that performed phototaxis, controlled by a simulated neural network; the robot's mechanical configuration and network controller were patterned after those thought to produce chemotaxis in the nematode worm, *C. elegans*. Webb and colleagues designed and tested robots based on phonotaxis in crickets [30, 31], and chemotaxis in ants [26, 32]. Kodjabachian and Meyer [19] evolved an artificial insect to follow an odor gradient, using “simple geometry-based cellular encoding” to evolve developmental programs that generated complex recurrent neural networks.

Other approaches are based on the idea of tactic response, but emulate the general behavior of taxis instead of a specific biological instance of it. Christensen and Dorigo [12] evolved neural network controllers for cooperative mobile robots that integrated phototaxis and hole-avoidance behavior. Floreano and Mondada [17] combined neural network and evolutionary computation methods to evolve a robot that could navigate to a battery charger; the charger's location was indicated by the presence of a light tower above the battery area and an area of black floor covering the charger area. Using a genetic algorithm approach, Beer and Gallagher [6] evolved a continuous-time recurrent neural network (CTRNN) that allowed an artificial agent to perform chemotaxis, moving from arbitrary initial locations and orientations to a “food patch” at the center of a square arena. Watson *et al.* [29] evolved neural network controllers for robots to perform Braitenberg-like phototaxis [9], moving to a lamp in the center of a rectangular pen.

Taking their inspiration from a single cell, Onsum and Arkin [24] developed a collaborative control algorithm based on the behavior of a human neutrophil (white blood cell). The algorithm controlled an autonomous robot tracking a target by averaging the outputs of individual sensors located on the robot's periphery, sending the resulting single signal

to the robot’s actuators. Dhariwal *et al.* [15] developed a robotic implementation of a “biased random walk” as observed in the motion of *E. coli*, conducting experiments both in simulation and on a robot. They compared their biased random walk approach to traditional gradient descent methods, and found that, although gradient descent was faster, the biased random walk performed better with multiple and dissipative sources, and provided better boundary coverage.

Some computational systems are concerned primarily with modeling or exploring the underlying biology. Eyyiurekli *et al.* [16] developed a software system that simulates chemotaxis based cell aggregation in 2D, and implements cell behaviors such as chemical diffusion and detection, motility, proliferation, adhesion, and cell life cycle stages. The emergence and evolution of signal transduction networks was explored by Soyer *et al.* [28] using a generic signal transduction model to evolve the chemotactic response *in silico*.

2. METHODS

2.1 Avida: Overview

The Avida software system [20, 23] is a widely used digital evolution platform. In digital evolution [1], a form of evolutionary computation, a population of computer programs—“digital organisms”—is placed in a computational environment. The digital organisms self-replicate, mutate, and compete, so the processes occurring in Avida are an *instance* of evolution and not a *simulation* [14]. Digital evolution can be used both to provide a better understanding of biological processes and to apply lessons learned from biology to computational and engineering problems.

Avida constructs a discrete two-dimensional world as a grid of cells. Avida places a population of individual digital organisms into this world, with at most one organism per grid cell. An individual organism consists of a “genome” (a circular set of assembly-like instructions) and a virtual CPU. The CPU contains three general purpose registers, two heads (FLOW, used as a target for jumps, and IP, used as an instruction pointer), and two stacks. Execution of the instructions in the organism’s genome acts upon the elements of the virtual CPU, and instruction execution incurs a cost, in virtual CPU cycles. Performing logic operations, gathering information or resources from the environment, or changing orientation is accomplished by execution of Avida instructions. The instruction set is easily extensible, allowing for great flexibility in adding new instructions to expand the system’s capabilities.

Replication in Avida occurs by copying the parent organism’s genome into a block of memory that will become the offspring’s genome. The copying is imperfect, however, leading to differences between the parent’s genome and that of its offspring. These differences, or *mutations*, can take the form of an instruction changing, or an instruction’s insertion or deletion. The Avida instruction set itself is robust, so that any program will be syntactically correct even with the occurrence of mutations [22]. When an organism replicates, its offspring is placed into a random grid cell, terminating any organism that was previously occupying the grid cell. In this way, the organisms compete for the limited space in the set of grid cells, and organisms that are able to replicate more quickly will have a greater proportion of descendants within the population. Organisms also accumulate merit bonuses by performing tasks that are specified by the user.

Merit is used to determine fitness; organisms with higher fitness are allotted more virtual CPU cycles in an update, allowing them to execute more quickly and replicate sooner than organisms with lower fitness. Fitness in Avida is, then, implicit rather than calculated by an explicit fitness function, as is seen in many evolutionary computation methods.

2.2 Motility and taxis in Avida

Until recently, Avida organisms (or Avidians) did not have the ability to move: an organism spent its entire lifetime in the same grid cell. Each Avidian has a facing, meaning the direction in which it is oriented. Facings for Avidians are discrete, in multiples of 45 degrees, as in the eight main compass directions (north, northeast, east, *etc.*). Some instructions were already present that allow an organism to turn right or left in different amounts. Several changes were made to the Avida instruction set in order to (1) implement fundamental motility, (2) create a simplified environment with an idealized gradient, (3) provide for sensing the gradient concentration, both with and without hand-coded memory supplied by the instruction, and (4) test different rotation strategies for gradient following.

2.2.1 Fundamental motility

The initial implementation of motility for Avidians was based on the tumble-and-run chemotaxis behavior as observed in bacteria such as *E. coli*, described earlier. Typically, an Avida run results in all grid cells being occupied by an Avidian; to allow room for movement to occur, a population cap was implemented that limits the total population size to a user-defined number of organisms. When the cap is reached, organisms are “killed” at random. We added two new instructions to provide motility itself, MOVE (*move*) and TUMBLE (*tumble*). The *move* instruction moves the organism one grid cell in the direction that the organism is currently facing. If there is already an organism in the destination grid cell, the two organisms exchange places; the displaced organism is completely passive. The *tumble* instruction selects a facing at random and reorients the organism to that facing. Seven of the eight possible facings are available to be selected; it is not possible for the organism to remain in the same orientation following the execution of a *tumble*.

2.2.2 Idealized gradient, sensing, and rotation

Idealized gradient. Living organisms are surrounded by gradients that involve different sensory modalities, and that are subject to differing types of dispersal (*e.g.*, diffusion, turbulence) depending on the environment and the often complex dynamics within it. This complexity can produce behavior in organisms that is difficult to interpret and understand. Controlling for those environmental factors has been an important part of the study of microorganisms (see, *e.g.*, [4, 7]). Within the digital world of Avida, those potentially confounding factors can be controlled, and environments constrained, in order to better illuminate the behavior of the digital organisms.

The notion behind the idealized gradient is that a simple, easily calculated quantity can be used to establish the gradient within the environment, removing the need for complex artificial physics. New sensing instructions, easily added to the Avida instruction set, function as “black boxes:” the or-

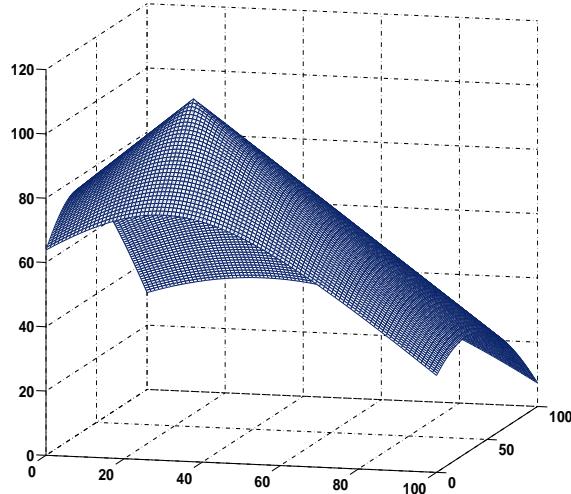


Figure 1: Example of distance as an idealized gradient. The “concentration” source location is at the top of the hill. Note that it is not necessary for the peak to be in the center of the environment.

ganism itself has access only to information that is placed in its virtual CPU registers, not to the information that is used inside the instruction to produce that information. All input values and calculation details are hidden from the organism inside the instruction, just as our own brain’s computation is hidden from us. For example, humans do not have cognitive access to either the physical properties of the input from a light stimulus, or to the visual system’s step-by-step computational processing of that input. We have access only to the output of the computation, *i.e.*, our perception of the stimulus. Similarly, the Avidian has access only to the output of its sensing instructions. This detail makes it possible to implement instructions that permit the Avidians to “sense” any salient feature or quantity within the environment.

In the current study, the Euclidean distance between two grid cells, the organism’s position and a “target” location, provided the gradient. “Target” is used here in the sense of a source of a sensory stimulus. The distance between the current and target locations is analogous to the concentration of a particular attractant that the organism will move toward (Figure 1). The fact that distance is a high-level construct makes no difference in this context: the Avidian “senses” the distance as a simple integer value. This sensed “concentration” is placed into a virtual register by the associated instruction, making that value available for processing. As previously discussed, the (x, y) coordinates of the organism’s and target’s positions and all the other calculation details are hidden from the organism inside the instruction’s implementation; the only information available to the organism is the current concentration (distance to target).

Sensing. At birth, each Avidian was given a target grid cell in a random location at least a user-specified minimum distance away. Since the target grid cell is analogous to an

attractant source, the instruction that was implemented to “sense” the distance between the organism and its target grid cell is equivalent to an idealized sensory system. In biological systems, both the sensory system and the organism’s interest in the particular attractant would have evolved. In digital evolution, it is reasonable to allow some capabilities to be provided from the start, if those capabilities are not the central concern of the investigation. Since the current study focuses on motility and tactic response, it is reasonable to allow the sensory system and the effect of the attractant to be available at the offset, since the evolution of those mechanisms is not within the scope of the current work.

We added a new instruction to the Avida instruction set, SENSE-TARGET-DISTANCE ($s-t$), that “senses” the current distance between the organism and its target, determines if any merit bonus is to be given to the organism, and places distance information into the organism’s registers. The organism receives a merit bonus for executing the instruction. The *sense-target* instruction reward structure is aimed at providing a large incentive for executing the sense instruction. An organism’s total merit bonus reflects how much of the distance between its initial position and its target the organism travels during its lifetime. An organism receives the maximum merit bonus if it reaches its target exactly; lesser distances covered are rewarded proportionally. Merit bonuses accrue based on an organism’s improving its situation: the organism receives the bonus if it is now closer to its target than during the previous execution of $s-t$. The reward structure does not give larger rewards to organisms that start farther away from their targets; the distance between the initial position and the target is normalized. Organisms are not “punished” for poor behavior, *i.e.*, there is no penalty for movement away from the target. The sensing instruction is not directly coupled to the movement instructions, nor are the organisms compelled to utilize the sensory information. The reward structure encourages evolution to make use of movement, since the merit bonus increases only when an organism is closer to its target than when the instruction was last executed, and the fact that the sensory information is completely reliable, *i.e.*, there is no noise, encourages the organisms to evolve tactic behavior. The instruction also provides for a small hand-coded “memory:” the instruction loads the current distance into one register and the previous distance into a second register, making the two values available for comparison.

We also implemented a slightly different version of the sensing instruction that did not provide hand-coded memory. The SENSE-TARGET-INSTANT ($s-t-i$) instruction is identical to $s-t$ except that $s-t-i$ does not place the previous distance value into a register, and so provides only the “instantaneous,” or current, sense value. The result is that the Avidian does not automatically have sensed values from two different times available for comparison. Organisms using this instruction rather than the original $s-t$ must evolve a rudimentary memory process: due to the random placement of targets relative to organisms, memory is required for an Avidian to move closer to its target and accumulate a larger merit bonus.

2.2.3 Rotation strategies

As a first demonstration of how motility in Avida can be used, we performed experiments that compared the efficacy of two different rotation or orientation strategies in the

Table 1: New Avida instructions used for experiments.

Experiment	Instructions used
Taxis, hand-coded memory	<i>sense-target, move,</i> <i>tumble</i>
Taxis, evolved memory	<i>sense-target-instant,</i> <i>move, tumble</i>
Rotation strategy, random	<i>sense-target, move,</i> <i>tumble-and-move</i>
Rotation strategy, systematic	<i>sense-target, move,</i> <i>rotate-right-one-and-move</i>

context of simple gradient following. The effectiveness of the new *tumble* instruction was compared to that of an existing rotation instruction, ROTATE-RIGHT-ONE (*r-r-1*), that changes the organism’s facing by one 45 degree turn to the right (clockwise). These two strategies allow comparison between a random strategy (*tumble*) and a systematic strategy (*r-r-1*). To minimize the possibility of multiple turns occurring before the execution of a *move*, two specialized instructions were implemented, based on the existing rotation instructions. These instructions combined a turn, either a *tumble* or a *r-r-1*, with a *move* immediately following. TUMBLE-AND-MOVE (*t-m*) combined the random *tumble* and an accompanying *move*, and ROTATE-RIGHT-ONE-AND-MOVE (*r-r-1-m*) paired a single rotation of 45 degrees to the right with an immediate *move*. Compelling an organism to move immediately after a turn prevented the organisms from executing multiple consecutive turns, thus preserving the distinction between the random and systematic rotation schemes.

3. EXPERIMENTS AND RESULTS

3.1 Experimental design

Each Avida experiment seeded the population with a simple self-replicating organism, *i.e.*, an organism with only the capability to replicate. This seed organism’s genome consists of 100 instructions, composed of a short copy loop and a large number of no-op instructions. Other instructions can appear only through mutations. All experiments used the default Avida mutation rates, with a 0.0075 copy-mutation probability per copied instruction, and insertion and deletion mutation probabilities of 0.05 per divide (overall 0.085 genomic mutation rate for a length-100 organism) [23].

Each population was placed into a 100×100 cell bounded grid, with a population cap of 1000 organisms. The sparse population served to minimize organism interactions in movement, since the grid was only 10% full. Experiments ran for 100,000 updates (approximately 10,000-15,000 generations), with 50 (taxis experiments) or 100 (rotation strategies experiments) replicates. The experiments utilized four different instruction sets, each consisting of the 26-instruction default Avida instruction set [23] and the sense, movement, and rotation instructions that were appropriate to the specific treatment. Table 1 summarizes the sensing and movement instructions used for each experimental treatment.

The primary performance metric is the *best distance ratio*. The ratio is computed as $1 - d_b/d_i$, where d_b is the best distance to target (the distance value of the organism’s closest approach to its target), and d_i is the initial distance to target (the distance between the target and the organism

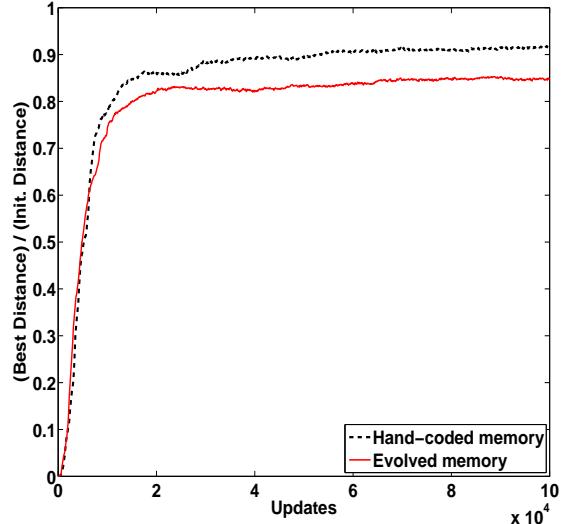


Figure 2: Average best distance ratio for experiments using *sense-target* instruction (both the current and the previous sensed value are given to the organism, but comparisons must be performed in addition to this instruction) and *sense-target-instant* (the organism must remember previously sensed values and perform comparisons in addition to this instruction). The average represents all organisms in the population at each update (total of approximately 1,000,000-2,000,000 total organisms per run), averaged over all 50 replicates of each treatment.

at birth). Statistics track and store the best distance ratio for each Avida organism, and compute the population’s average best distance ratio for each update. The best distance ratio statistic actually reports the average best distance ratio of the parents of the current population; this technique serves to reduce possible biasing caused by such factors as the proportion of “old” and “young” organisms in the current population.

3.2 Results

3.2.1 Taxis with and without hand-coded memory

As illustrated in Figure 2, tactic behavior evolved successfully, both with and without the use of hand-coded memory. The behavior emerged quite quickly, within the first one to two thousand updates. The speed with which motility emerged is demonstrated by the average best distance ratios over all 100,000 updates for the two treatments, 0.8396 for taxis with hand-coded memory and 0.7910 for taxis without hand-coded (*i.e.*, with evolved) memory.

The average best distance ratio for the last 50,000 updates shows the overall success of the evolved taxis strategies, 0.9028 using hand-coded memory, and 0.8480 with evolved memory. These highly successful strategies emerge quickly and become relatively stable within the first 50,000 updates. Although both treatments produce successful solutions, the use of hand-coded memory produces significantly better performance, as shown by a Mann-Whitney *U*-test comparing the average best distance ratios at the ends of each experimental run ($N = 50$, $p = 4.9286 \times 10^{-7}$).

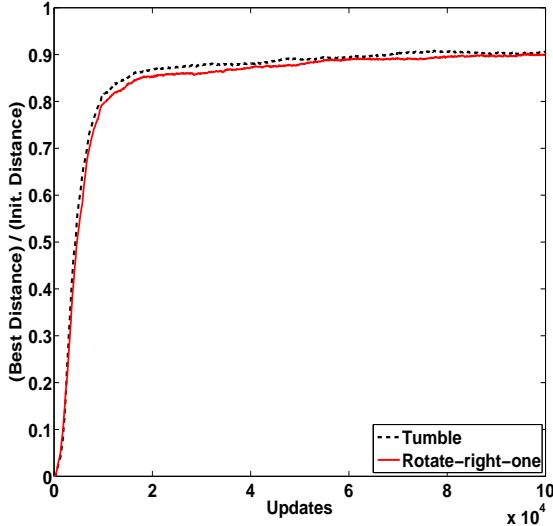


Figure 3: Average best distance ratio for *tumble-and-move* and *rotate-right-one-and-move* instructions. The figure shows the average best distance ratio over the entire population, for 100 replicates of each instruction set. Experiments ran for 100,000 updates (approx. 10,000-15,000 generations).

3.2.2 Varying rotation schemes

Both the random (*tumble*) and systematic (*r-r-1*) rotation strategies worked well in the experimental environment (Figure 3). The average best distance ratios for the last 50,000 updates were similar, 0.9004 using the *t-m* instruction, and 0.8926 using *r-r-1-m*. The *t-m* instruction appears to slightly outperform *r-r-1-m*, but a Mann-Whitney *U*-test shows the difference of the best distance ratios after 100,000 updates is not significant ($N = 100$, $p = 0.2856$).

4. DISCUSSION

The experiments that evolved simple taxis demonstrated two key points. First, the experiments served as proof of concept. The digital organisms will evolve motility and exploit that capability when placed in an environment where movement provides an adaptive advantage (Figure 4). Second, the experiments showed that hand-coded memory is useful, but not necessary for taxis to emerge. The hand-coded memory provided by the *s-t* instruction was part of the instruction itself. The instruction placed all the information the organism needed (*i.e.*, the current and previous sensed values) in the correct virtual registers, so the information was immediately available for further processing. Evolution still needed to find how to handle the information appropriately—with a comparison—but no additional mechanism for storing or retrieving the information was needed. This made the task of evolving the information processing portion of the tactic response much simpler, since it required mutating fewer instructions into the genome. In contrast, the *s-t-i* instruction provided no such extra help. The instruction placed the current sense information into a virtual register, but not the prior sense value. This one change compelled evolution to discover the mechanisms for storing

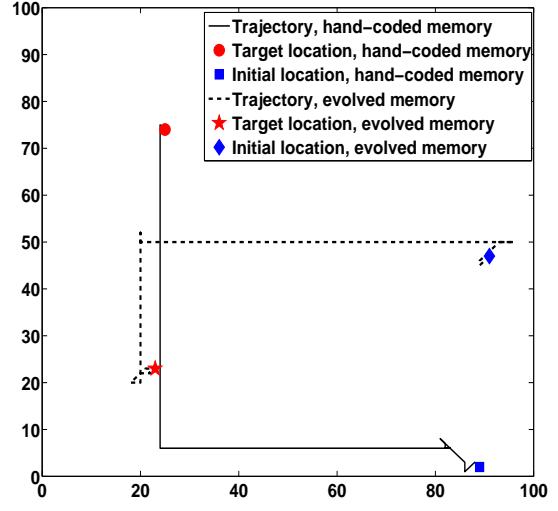
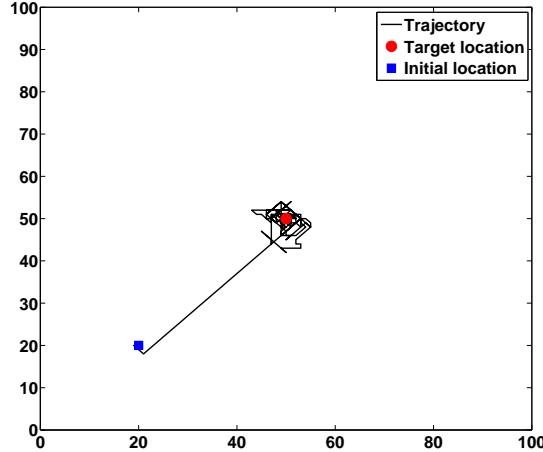


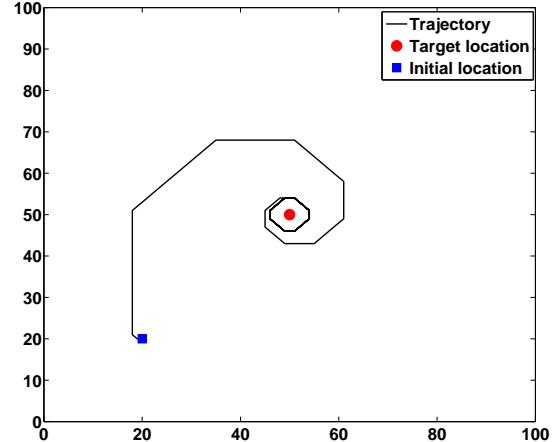
Figure 4: Sample trajectories of evolved organisms using *sense-target* instruction (both the current and the previous sensed value are given to the organism, but comparisons must be performed in addition to this instruction) and *sense-target-instant* (the organism must remember previously sensed values and perform comparisons in addition to this instruction). The trajectories suggest that the organisms are using the sensed information to track to their target locations.

and retrieving previously sensed information by changing the genome. This was clearly more difficult than the other condition, but it is exciting to see how evolution was able to produce successful solutions in the absence of hand-coded memory.

The rotation strategies results show that, at least in the idealized, discrete environment of the current study, both the random *tumble* and systematic *r-r-1* rotation strategies are “good enough.” there is no significant difference between the performances of the two rotation schemes, as measured by the average best distance ratios of the populations. This result may initially seem surprising, since *tumble* is based on an existing, successful biological orientation strategy, and *r-r-1* is highly artificial and biologically implausible, due to its rigid constraints on both direction and degree of turn. Figure 5 illustrates representative trajectories using each of these two rotation strategies. The trajectory plots were generated using movement data collected from hand-coded organisms in Avida runs with no mutations. The *t-m* instruction, shown in Figure 5a, allows the organism to reach the actual target location. The last step to the goal, however, can be difficult: due to the random choice of the new orientation, *tumble* has only a 1/8 probability of generating the correct orientation to reach the goal on the next step. On the other hand, *r-r-1-m*, as shown in Figure 5b, can never reach the full distance to the goal, because of the constraint of turning only 45 degrees before taking a step. Without the ability to do two consecutive turns, the organism can, at best, reach the grid cells one step away from the target. These particular traits are a liability in the present context



(a) TUMBLE-AND-MOVE



(b) ROTATE-RIGHT-ONE-AND-MOVE

Figure 5: Representative trajectories for *tumble-and-move* and *rotate-right-one-and-move*. These trajectories were produced from movement data gathered in Avida runs using hand-coded organisms, with mutations turned off. For ease of comparison, the organisms were assigned the same initial and target locations.

only because the target location was implemented as a single point; a more realistic implementation would allow the target to cover a slightly larger area, for example a 2×2 square. The respective strategies' difficulties notwithstanding, the process of evolution found mechanisms that sufficed in the idealized world of the experiments. This is a simple demonstration of “satisficing” in the evolutionary process: given a set of constraints, evolution will discover solutions that work well enough in the given environment. The term “satisficing” was coined by Herbert Simon, and is used to refer to a decision-making strategy that attempts to meet criteria of adequacy rather than to find an optimal solution. This is related to the concept of bounded rationality, that takes into account the limitations of both knowledge and cognitive capacity in making decisions [27]. Originally used in the context of economics, the idea of satisficing is a natural fit with evolutionary processes. In biological evolution, there is not necessarily any additional advantage in discovering a solution that is more than adequate; moving toward a more “optimal” solution will occur either by chance mutations or in response to changing selection pressures.

5. CONCLUSIONS

The work presented here represents the initial steps in implementing motility in the Avida software system, and demonstrates that it is possible to evolve simple motility and related behavior. Future work will include implementation of additional infrastructure, some of which will be aimed at creating a richer, more complex environment where evolution can unfold. This will include moving from the discrete, grid-based topology to a continuous, 3D topology, and using more complex, less idealized environments to enable exploring the relationship between sensors, environments, and orientation strategies.

The fact that simple taxis evolved without the additional assistance of hand-coded memory is particularly tantalizing, suggesting that it may be possible to investigate aspects

of the emergence of memory during evolution using Avida. Memory is generally considered an important component of intelligence, so the evolution of memory will be of great interest in the context of emerging intelligent behavior.

The introduction of movement into Avida opens up many intriguing avenues of inquiry that were not previously possible using this software system. It is now possible to begin to use the Avida digital evolution platform to look at ongoing questions in biology that relate to movement, such as the evolution of efficient movement, foraging, and navigation. These movement and decision-making problems present great challenges, but also hold great potential for offering insights into the evolutionary emergence of intelligent behaviors, insights that can then be applied to computational problems.

6. ACKNOWLEDGMENTS

We wish to acknowledge the valuable contributions of Fred C. Dyer, of the Department of Zoology at Michigan State University, and the assistance of our colleagues at the Digital Evolution Laboratory at Michigan State University, including Brian Baer, Jeff Barrick, Benjamin Beckmann, David Bryson, and Michael Vo. This research was supported by a grant from the Cambridge Templeton Consortium, “Emerging Intelligence: Contingency, Convergence and Constraints in the Evolution of Intelligent Behavior,” and by NSF grants CCF-0523449 and EIA-219229.

7. REFERENCES

- [1] C. Adami, C. A. Ofria, and T. C. Collier. Evolution of biological complexity. *Proceedings of the National Academy of Science*, 97:4463–4468, 2000.
- [2] J. Adler. Chemotaxis in bacteria. *Science*, 153(3737):708–716, August 12 1966.
- [3] J. Adler. Chemotaxis in bacteria. *Annual Review of Biochemistry*, 44:341–356, 1975.

- [4] J. Adler. *Primitive sensory and communication systems: the taxes and tropisms of micro-organisms and cells*, chapter Chemotaxis in bacteria, pages 91–100. Academic Press, London, 1975.
- [5] J. Ayers, J. Witting, N. McGruer, C. Olcott, and D. Massa. Lobster robots. In T. Wu and K. N, editors, *Proceedings of the International Symposium on Aqua Biomechanisms*, Tokai University, 2000.
- [6] R. D. Beer and J. C. Gallagher. Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):91–122, 1992.
- [7] H. C. Berg. *E. coli in motion*. Springer, New York, 2004.
- [8] H. C. Berg and D. A. Brown. Chemotaxis in *Escherichia coli* analysed by three-dimensional tracking. *Nature*, 239:500–504, 27 October 1972.
- [9] V. Braatenberg. *Vehicles: experiments in synthetic psychology*. MIT Press, Cambridge, MA, 1984.
- [10] K. L. Briggman, H. D. I. Arbelan, and W. B. Kristan Jr. Optical imaging of neural populations during decision-making. *Science*, 307(5711):896–901, 2005.
- [11] M. J. Carlile. *Primitive sensory and communication systems: the taxes and tropisms of micro-organisms and cells*, chapter Taxes and tropisms: diversity, biological significance and evolution, pages 1–28. Academic Press, London, 1975.
- [12] A. L. Christensen and M. Dorigo. Evolving an integrated phototaxis and hole-avoidance behavior for a swarm-bot. In L. M. Rocha, L. S. Yaeger, M. A. Bedau, D. Floreano, R. L. Goldstone, and A. Vespignani, editors, *Proceedings of the 10th International Conference on the Simulation and Synthesis of Living Systems (Alife X)*, pages 248–254. MIT Press, Cambridge, MA, 2006.
- [13] J. D. E. Koshland. A model regulatory system: bacterial chemotaxis. *Physiological Reviews*, 59(4):811–862, October 1979.
- [14] D. C. Dennett. The new replicators. In M. Pagel, editor, *Encyclopedia of Evolution*, pages E83–E92. Oxford University Press, New York, 2002.
- [15] A. Dhariwal, G. Sukhatme, and A. A. Requicha. Bacterium-inspired robots for environmental monitoring. In *Proceedings of 2004 IEEE/RSJ International Conference on Robotics and Automation*, pages 1436–1443, New Orleans, LA, April 2004.
- [16] M. Eyyiurekli, P. I. Lelkes, and D. E. Breen. A computational system for investigating chemotaxis-based cell aggregation. In F. A. e Costa, L. M. Rocha, E. Costa, I. Harvey, and A. Coutinho, editors, *Advances in Artificial Life, 9th European Conference, ECAL 2007, Lisbon, Portugal, September 10-14, 2007, Proceedings ECAL*, number 4648 in Lecture Notes in Computer Science, pages 1034–1049. Springer, 2007.
- [17] D. Floreano and F. Mondada. Evolutionary neurocontrollers for autonomous mobile robots. *Neural Networks*, 11:1461–1478, 1998.
- [18] F. W. Grasso, T. R. Consi, D. C. Mountain, and J. Atema. Biomimetic lobster performs chemo-orientation in turbulence using a pair of spatially separated sensors: progress and challenges. *Robotics and Autonomous Systems*, 30:115–131, 2000.
- [19] J. Kodjabachian and J.-A. Meyer. Evolution and development of neural controllers for locomotion, gradient-following, and obstacle-avoidance in artificial insects. *IEEE Transactions on Neural Networks*, 9(5):796–812, 1998.
- [20] R. E. Lenski, C. Ofria, R. T. Pennock, and C. Adami. The evolutionary origin of complex features. *Nature*, 423:139–144, 2003.
- [21] T. M. Morse, S. R. Lockery, and T. C. Ferrée. Robust spatial navigation in a robot inspired by chemotaxis in *Caenorhabditis elegans*. *Adaptive Behavior*, 6:393–410, 1998.
- [22] C. Ofria, C. Adami, and T. C. Collier. Design of evolvable computer languages. *IEEE Transactions in Evolutionary Computation*, 17:528–532, 2002.
- [23] C. Ofria and C. O. Wilke. Avida: a software platform for research in computational evolutionary biology. In *Artificial Life 10*, pages 191–229, 1994.
- [24] M. D. Onsum and A. P. Arkin. Autonomous mobile robot control based on white blood cell chemotaxis. *Lecture Notes in Computer Science*, 3082:9–19, 2005.
- [25] G. W. Ordal. Bacterial chemotaxis: a primitive sensory system. *BioScience*, 30(6):408–411, 1980.
- [26] T. Sharpe and B. Webb. Simulated and situated models of chemical trail following in ants. In R. Pfeifer, B. Blumberg, J.-A. Meyer, and S. W. Wilson, editors, *From animals to animats 5: proceedings of the fifth international conference on simulation of adaptive behavior*, pages 195–204, Cambridge, MA, 1998. MIT Press.
- [27] H. A. Simon. *Models of man—social and rational*. John Wiley and Sons, New York, 1957.
- [28] O. S. Soyer, T. Pfeiffer, and S. Bonhoeffer. Simulating the evolution of signal transduction pathways. *Journal of Theoretical Biology*, 241:223–232, 2006.
- [29] R. A. Watson, S. G. Ficici, and J. B. Pollack. Embodied evolution: embodying an evolutionary algorithm in a population of robots. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 335–342, Mayflower Hotel, Washington D.C., USA, 6–9 1999. IEEE Press.
- [30] B. Webb. Robotic experiments in cricket phonotaxis. In D. Cliff, P. Husbands, J. Meyer, and S. Wilson, editors, *From animals to animats 3: proceedings of the fifth international conference on simulation of adaptive behavior*, Cambridge, MA, 1994. MIT Press.
- [31] B. Webb. Using robots to model animals: a cricket test. *Robotics and Autonomous Systems*, 16:117–134, 1995.
- [32] B. Webb. Robots, crickets and ants: models of neural control of chemotaxis and phonotaxis. *Neural Networks*, 11:1479–1496, 1998.