

# Directed Evolution of Communication and Cooperation in Digital Organisms

David B. Knoester, Philip K. McKinley, Benjamin Beckmann, and Charles Ofria

Department of Computer Science and Engineering  
Michigan State University  
East Lansing, Michigan 48824  
{dk, mckinley, beckma24, ofria}@cse.msu.edu

**Abstract.** This paper describes a study in the use of digital evolution to produce cooperative communication behavior in a population of digital organisms. The results demonstrate that digital evolution can produce organisms capable of distributed problem solving through interactions between members of the population and their environment. Specifically, the organisms cooperate to distribute among the population the largest value sensed from the environment. These digital organisms have no “built-in” ability to perform this task; each population begins with a single organism that has only the ability to self-replicate. Over thousands of generations, random mutations and natural selection produce an instruction sequence that realizes this behavior, despite continuous turnover in the population.

**Keywords:** digital evolution, communication, cooperative behavior, natural selection, mutation, autonomic computing, biologically-inspired computing.

## 1 Introduction

The increasing interaction between computing technology and the physical world requires that systems with different characteristics and capabilities be able to reliably communicate, regardless of changing environmental conditions [1]. Similar to how living organisms have evolved remarkable methods (audible, visual, stigmergic) for communication, we can use *digital evolution* [2] to evolve communication strategies for distributed computing systems. By utilizing an evolutionary process that incorporates many of the hazards to communication (packet loss, node failure), solutions that would not otherwise be apparent to human designers may be discovered.

Our work uses the AVIDA platform for digital evolution [2] to investigate the evolution of cooperative communication behavior. In AVIDA, a population of self-replicating computer programs exists in a user-defined computational environment and is subject to instruction-level mutations and natural selection. Over thousands of generations, these “digital organisms” can adapt to and even thrive under extremely dynamic and adverse conditions. AVIDA has previously been used to conduct research in the evolution of biocomplexity, with an emphasis on the evolutionary design process in nature [3, 4]. However, digital evolution can also be used to address complex problems in science and engineering [5, 6], often revealing unexpected and clever solutions.

Biologically-inspired approaches and evolutionary computation have been applied to a variety of cooperative communication problems. Examples include mimicking the social behavior of insect colonies in robotic foraging [7] and using chemotaxis to facilitate robust network routing [8]. In addition, a variety of studies have been conducted to better understand the evolution of cooperation and communication. Examples include the evolution of a common vocabulary [9, 10], using the Prisoner's Dilemma to examine the evolution of cooperation [11], the effect of communication and indirect reciprocity on the evolution of cooperative strategies [12–14], and how information flow between agents is shaped by interaction with the environment [15, 16].

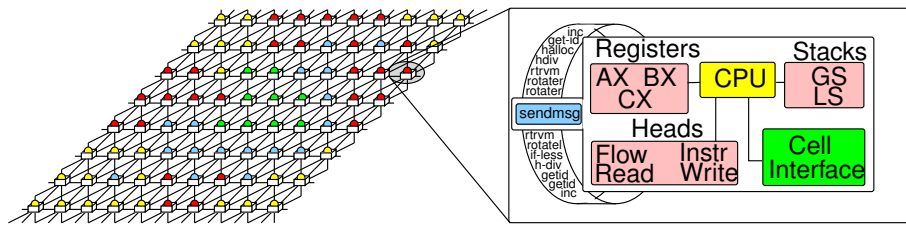
This study focuses on the evolution of a distributed problem solving task [17], specifically, where the population must determine the largest value sensed by any individual. Such behavior could provide a means for a population to perform leader election [18], or could be used to obtain the maximum sensed value in a wireless sensor network [19]. Our results show that digital evolution can produce this behavior, and therefore has promise as a tool to be used in the design of future distributed computing systems. Like natural organisms, those systems will need to adapt to the environment, self-heal, and evade attackers. After reviewing the AVIDA platform, we describe our experiments, present results, and analyze the dominant genome of a population that evolved the desired behavior. Finally, we present conclusions and discuss future work.

## 2 AVIDA Background

Figure 1 depicts an AVIDA population and the structure of an individual organism. Each digital organism comprises a circular list of instructions (its genome) and a virtual CPU, and “lives” in a common virtual environment. AVIDA instructions are similar in appearance to traditional assembly language instructions. They enable an organism to perform simple mathematical operations, such as addition, multiplication, and bit-shifts, as well as interact with the organism's environment, for example, by sending a message to a neighboring organism, or outputting a number to the environment. Instructions are executed by the organism's virtual CPU; the one used here contains three registers, two stacks, and four *heads*, which are similar to program and stack pointers [2].

The AVIDA environment comprises a number of *cells*, each of which can contain at most one organism; organisms cannot live outside of cells. Each cell has a circular list of directed *connections* to neighboring cells. These connections define the topology of the environment. A single connection, the *facing*, defines the orientation of the resident organism. The facing of a cell may be sensed and manipulated by the resident organism using the GET-FACING and ROTATE- $\{L,R\}$  instructions, respectively. Finally, each cell in the environment has an associated identifier, a unique random 32-bit integer, termed the *cell-ID*. A resident organism may obtain its cell-ID via the GET-ID instruction. Organisms in AVIDA can communicate with each other by sending and receiving messages in the direction currently faced. If the sending organism is facing a neighboring organism, the message is deposited in that neighbor's inbox. If the sender was facing an empty cell, the message is lost. The recipient of the message must execute a RETRIEVE-MSG instruction to extract the message from its inbox. Organisms are not able to determine if they are facing an occupied cell, nor do we provide an explicit mechanism for them to identify neighbors (though it may be evolved).

A population starts with a single organism that is capable only of replication, and different genomes are produced through random mutations that occur during replication. The first step in replication is for the parent to allocate space for the offspring's genome. The parent then executes its "copy-loop," where instructions are copied individually from the parent's genome to the offspring's. Finally, the parent organism executes an H-DIVIDE instruction, creating the offspring. Each time an instruction is copied, a mutation may be introduced according to a predefined probability. These mutations may take the form of a replacement (substituting a random instruction for the one copied), an insertion (inserting an additional, random instruction into the offspring's genome), or a deletion (removing the copied instruction from the offspring's genome). When an organism replicates, a target cell that will house the new organism is selected from the environment. Different models to select this target cell are available, including MASS-ACTION (select at random from among all cells) and NEIGHBORHOOD (select from cells adjacent to the parent), among others. In every case, an organism that is already present in the target cell is *replaced* (killed and overwritten) by the offspring.



**Fig. 1.** An AVIDA population (left), and the structure of an individual organism (right).

During an AVIDA experiment, the *merit* of a given digital organism determines how many instructions its virtual CPU is allowed to execute in relation to other organisms, similar to a priority-based scheduling algorithm. Since digital organisms are self-replicating, a higher merit results in an organism that replicates more frequently, spreading throughout and eventually dominating the population. Unlike fitness in genetic programming, merit in AVIDA is not evaluated only at discrete time intervals, but rather updated asynchronously based upon performed *tasks*. Tasks are designed by the user and are used to reward desirable behavior (they may also punish undesirable behavior), thereby driving natural selection. Tasks are defined in terms of an organism's externally visible behavior (its phenotype, for example, messages that are sent), rather than in terms of CPU-level actions. This approach is intended to allow maximum flexibility in the evolution of a solution for a particular task. The evolved solution might not be optimal when considering the task in isolation, but it is likely to have other properties that made it well-suited for its environment – resilience to mutation, for example. Multiple tasks can reward organisms for exhibiting complex behaviors. For example, one task may reward organisms for sending a message, while another may reward for a specific message payload. Rewards for performing multiple tasks are, by default, multiplicative.

### 3 Experiments and Results

We present three different sets of experiments. Each uses different combinations of AVIDA tasks, however all are focused on evolving the same behavior: proliferation of messages that carry the largest sensed value. For this study we use cell-IDs for the sensed values, thus the desired behavior is that all organisms send messages containing the largest cell-ID. Taking into account population turnover and mutations, we consider the solution to have been found when 95% of messages carry the largest cell-ID. The first set of experiments investigates the basic communication capabilities of digital organisms, focusing on the evolution of message filtering. The second set introduces a penalty, where each time that a digital organism sends a message that does not contain a cell-ID, the sender’s merit is reduced. Finally, the third set of experiments investigates the ability of the population to recover from resetting the largest cell-ID. We note that organisms do not have an inherent ability to identify messages that contain a cell-ID; both the messaging behavior, as well as the grammar, must be evolved.

**Experimental setup.** For this study we configured AVIDA as follows. The environment comprises 3600 cells in a  $60 \times 60$  torus. Experiments are run for 100,000 *updates*, a standard unit of time in AVIDA; an update averages 30 virtual CPU instructions per organism. The copy mutation rate is set to 0.75%, while the insertion and deletion mutation rates are set to 0.5%; these parameters correspond to the default AVIDA configuration. We developed a set of tasks, summarized in Table 1, to reward organisms for various communication behaviors. To account for the stochastic nature of evolution, 20 separate runs of AVIDA were performed for each experiment.

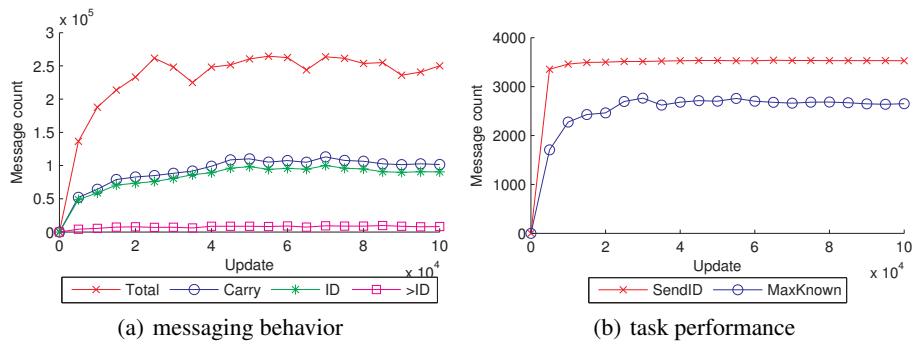
**Table 1.** Descriptions of the AVIDA tasks developed for this study.

Task Name	Description
SEND-SELF	Rewards sending a message containing the sender’s cell-ID.
SEND-ID	Rewards sending a message containing <i>any</i> cell-ID.
MAX-KNOWN	Rewards sending a message containing the largest value known, defined as $Max(self, Max(msg_0, \dots, msg_n))$ , where $\{msg_0, \dots, msg_n\}$ is the set of all messages received by that organism. The sender must have received at least one message prior to being rewarded for performing this task.
SEND-NON-ID	Penalizes the sender of a message that does not carry a cell-ID.

#### 3.1 Filtering Messages

In the first set of experiments, we tested the hypothesis that rewarding organisms for sending messages containing cell-IDs, where the cell-ID carried is greater than the organism’s own cell-ID, will eventually result in all messages in the population carrying the largest cell-ID. Experiments were conducted using different combinations of the tasks defined in Table 1. In every case, experiments that used the MAX-KNOWN task produced organisms sending messages containing cell-IDs greater than their own. However, none of these experiments resulted in the proliferation of the largest cell-ID.

Figure 2(a) depicts average messaging behavior for an experiment uses the MAX-KNOWN and SEND-ID tasks. Five different values are plotted over the previous 100 updates: Total, the total number of messages sent; Carry, the number of messages sent that carry a cell-ID; ID, the number of messages sent that carry the sender’s cell-ID; and >ID, the number of messages sent that carry a cell-ID greater than the sender’s. Here we see that more than half of all messages sent do not carry an ID, indicated by the difference between Total and Carry - these are “junk” messages, produced when organisms send values that are easy to calculate or when a SEND-MSG instruction has been mutated into the genome. We also see that greater than 75% of ID-carrying messages contain the sender’s cell-ID. Finally, very few messages contain an ID that is greater than the sender’s cell-ID. Figure 2(b) shows the average number of *organisms* that performed the MAX-KNOWN and SEND-ID tasks during the same AVIDA runs. Here we see that when MAX-KNOWN task is used in combination with the SEND-ID task, not only do all (allowing for genetic drift) organisms perform the SEND-ID task, but a large number of organisms (2600) also perform the MAX-KNOWN task. We note that organisms can perform the MAX-KNOWN task by sending their own ID once they have received any message carrying a smaller value.



**Fig. 2.** Data filtering with MAX-KNOWN and SEND-ID tasks; average of 20 runs.

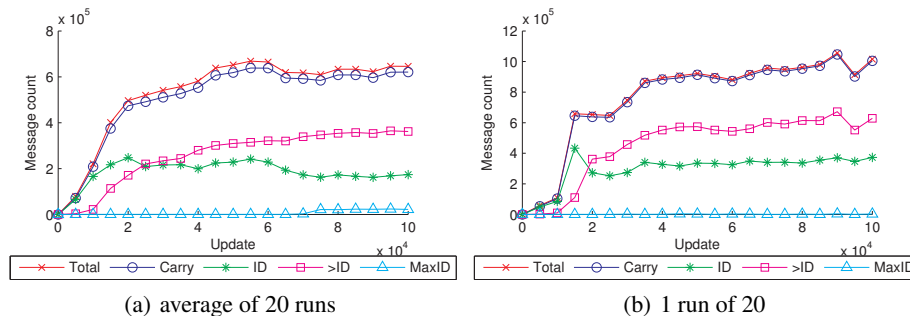
### 3.2 Encouraging ID-Carrying Messages

Our next experiments investigated ways to reduce the number of “junk” messages being sent, under the supposition that the large number of non-ID carrying messages might be preventing the population from determining the largest cell-ID. We tried two different approaches, one where we actively penalized organisms for sending junk messages, and another where we increased the cost (in virtual CPU cycles) of the SEND-MSG instruction. Both of these approaches resulted in the desired behavior, with the penalty evolving a solution more quickly than the additional cost approach. Here we discuss only the former; details of the cost experiments can be found in a technical report [20].

In this experiment, a task, SEND-NON-ID, was defined such that the sender of a message that does not carry a cell-ID is docked 75% of its merit. The SEND-NON-ID task is similar to an unseen predator, or hostile and unpredictable environment, in biological systems. We tried two different configurations with SEND-NON-ID, one that included

MAX-KNOWN and SEND-ID, and another that included MAX-KNOWN and SEND-SELF. Initial experiments that used the SEND-NON-ID penalty performed similarly to those described in Section 3.1. However, when we also changed the replacement strategy from MASS-ACTION to NEIGHBORHOOD performance improved dramatically. The reason for this improvement is related to *kin selection* [6], which occurs when parent and offspring work together on cooperative tasks. In this case, parent and offspring are genetically similar, and thus likely to cooperate on the rewarded tasks, while avoiding the SEND-NON-ID penalty. We note that NEIGHBORHOOD replacement alone, without using either a penalty or cost, did not achieve the desired behavior.

Figure 3 shows messaging behavior using the tasks MAX-KNOWN, SEND-ID, and the penalty SEND-NON-ID. In addition to the values plotted in Figure 2(a), we also plot MaxID, the number of messages sent that carry the largest cell-ID in the population. Figure 3(a) shows the average messaging behavior of 20 different runs, and Figure 3(b) is a detail of a single run selected to show the improvement in the types of messages present in the population. Here we see that the number of junk messages has been dramatically reduced, and that the number of messages containing IDs greater than that of the sender is increasing, although slowly. Still, very few messages contain the largest cell-ID.



**Fig. 3.** Messaging behavior with a penalty.

Figure 4 shows messaging behavior using the tasks MAX-KNOWN, SEND-SELF, and the penalty SEND-NON-ID. Figure 4(a) shows the average behavior of 20 different runs. For the first time, we see evidence of the convergence of message types, where the number of messages carrying an ID greater than the sender's approaches the total number of messages sent. However, we observe that due to the SEND-SELF task, each organism sends its own ID at least once. We also see a significant number of messages that carry the largest cell-ID. Figure 4(b) plots details of a particular run where nearly all messages contain cell-IDs greater than that of the sender. Moreover, those containing the largest cell-ID (MaxID) represent 98.3% of all sent messages. It is this behavior, where nearly all sent messages converge to the largest cell-ID, that we sought. We note that the drop in total number of sent messages corresponds to the evolution of filtering; a genome that exhibits this same behavior is described in Section 3.3.

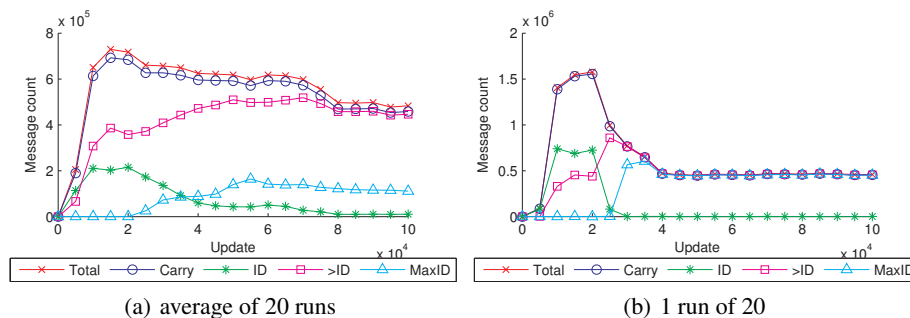


Fig. 4. Messaging behavior with a penalty, using SEND-SELF.

### 3.3 Recovery from ID Reset

Having determined that populations of digital organisms could cooperate to determine the largest cell-ID, we next investigated whether the population could react to a change in that ID. Using the penalty-based task configuration described earlier, we added an event, RESET-ID, that when executed, resets the largest cell-ID in the population to a smaller random value. Figure 5 shows messaging behavior with the RESET-ID event configured to occur at update 50,000, and using tasks MAX-KNOWN, SEND-SELF, and SEND-NON-ID. Figure 5 shows the average of 20 different runs, while Figure 5(b) shows the details of a single run that recovered from changing the largest cell-ID. In these figures, we see that the populations are not only able to recover from the change to the largest cell-ID, but that they exceed their pre-reset levels within 10,000 updates.

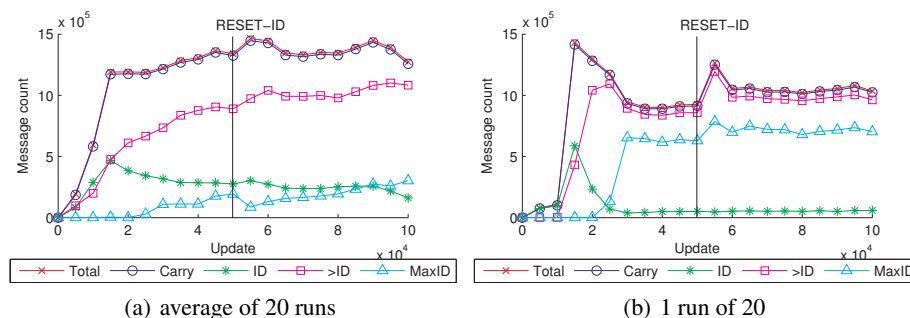
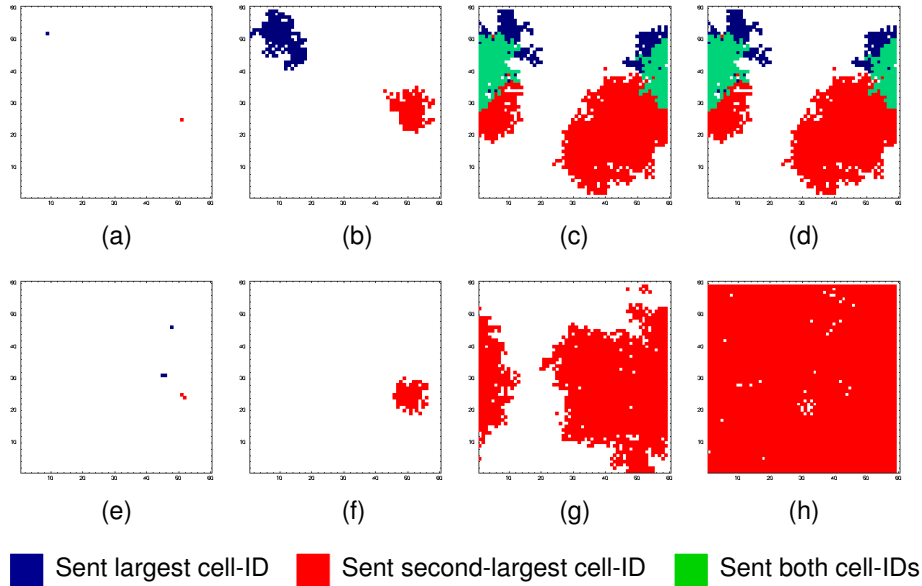


Fig. 5. Recovering from a change to the largest cell-ID.

Figure 6 shows keys stages of messaging behavior from Figure 5(b). (The full video of this run, with additional description, is available at the URL: <http://www.cse.msu.edu/~mckinley/avida>.) Figure 6 comprises snapshots of the population during the evolution process. Each snapshot identifies which organisms are sending the largest cell-ID, which are sending the second-largest cell-ID, and which send both during their lifetime. By frame (d) nearly all organisms are sending messages carrying the largest cell-ID; a few organisms near the cell with the second-largest ID are sending both IDs. The largest cell's ID is reset just prior to frame (e). As shown, the transmis-

sion of the (old) largest cell-ID dies out quickly. The population, however, is able to recover and quickly proliferate messages that carry the new largest cell-ID.

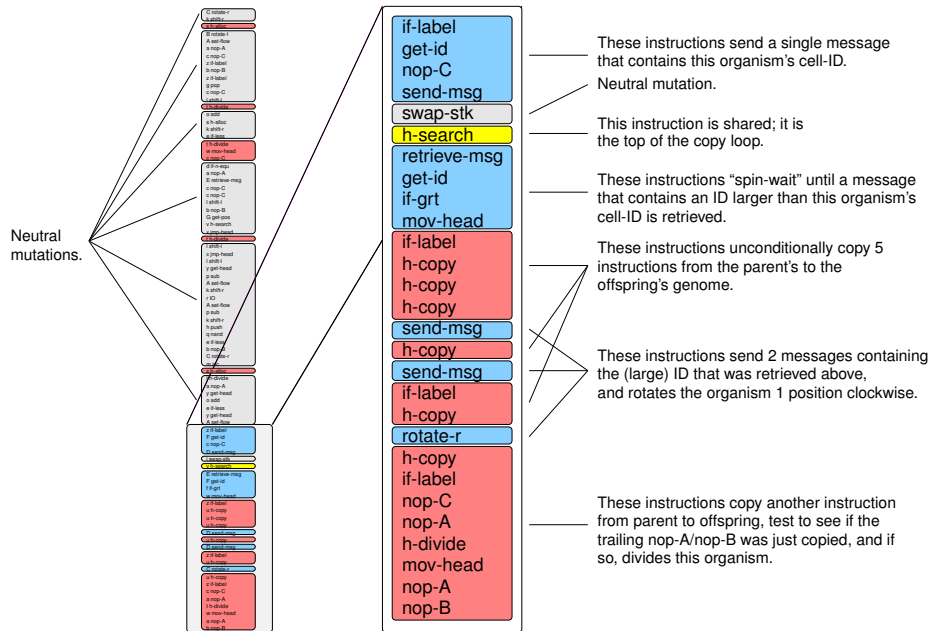


**Fig. 6.** Eight frames excerpted from an AVIDA trace, demonstrating the evolution of distributed problem solving.

Figure 7 shows the genome responsible for the behavior in Figures 5 and 6. In this figure we have identified neutral mutations as well as those parts of the genome that are relevant to determining the largest cell-ID and the replication cycle. This particular genome comprises 84 instructions, of which 12 are responsible for the desired behavior, 22 are responsible for the organism’s replication cycle, 1 instruction is shared, and 51 instructions, or 61% of the genome, are neutral mutations. An interesting feature of this particular genome is that its replication is dependent upon it receiving a message that carries a cell-ID larger than its own. In other words, organisms with this genome have evolved to the point where they depend upon the behavior of other organisms for their very survival. Specifically, if these organisms do not receive a message that has a data field larger than their own cell-ID, *they will not reproduce*.

It should be noted that one of the forces at work in evolving this behavior is the natural selection of organisms that do *not* perform the SEND-NON-ID task. As soon as the RESET-ID event is triggered, any organism that sends the original largest cell-ID is subject to the penalty for sending a junk message. Even in the absence of an explicit penalty, organisms that send the original largest cell-ID would still not receive the reward for the MAX-KNOWN task. It is these *selective pressures* that are primarily responsible for the distribution of the largest cell-ID. Moreover, an organism cannot be rewarded for sending a message containing the new largest cell-ID without first having been sent that ID in a message (unless, of course, the organism lives in that cell). In other words, to survive a change in the largest ID, the organisms depend on cooperation.





**Fig. 7.** Dominant genome responsible for the proliferation of messages that carry the largest cell-ID.

## 4 Conclusions and Future Work

We have demonstrated that digital evolution can produce populations capable of distributed problem solving, specifically distributing the largest cell-ID among the population. Furthermore, we have shown that in the presence of selective pressures, populations of organisms are able to recover from changes in their environment, and that this behavior emerges from simple localized interactions between neighboring organisms.

Our ongoing and future investigations include the following. First, we have used group selection to evolve organisms that perform leader election by identifying unique characteristics of individuals [21]. Second, we have used AVIDA to evolve organisms that generate UML state diagrams for dynamically adaptive systems. Third, we are using AVIDA to study the evolution of other distributed operations, such as data gathering, which can be applied to wireless sensor networks. Fourth, to study evolution in individuals capable of movement, such as mobile robotic agents, we have recently developed an instruction set that includes simple motor control primitives and sensors. We expect to use this platform to evolve individuals that use these new features in order to traverse obstacle courses, elude predators, and catch moving targets.

*Further Information* Papers on digital evolution and the AVIDA software are available at <http://devolab.cse.msu.edu>. Information on evolving cooperative behavior can be found at <http://www.cse.msu.edu/~mckinley/avida>.

*Acknowledgments* The authors gratefully acknowledge the contributions of the following individuals to this work: Ben Beckmann, Jeffrey Clune, Sherri Goings, David Bryson, Richard Lenski, Heather Goldsby, and Betty Cheng. This work was supported in part by the U.S. Department of the Navy, Office of Naval Research under Grant No. N00014-01-1-0744, U.S. National Science Foundation grants EIA-0130724, ITR-0313142, and CCF 0523449, and a Quality Fund Concept grant from Michigan State University.

## References

1. McKinley, P.K., Sadjadi, S.M., Kasten, E.P., Cheng, B.H.C.: Composing adaptive software. *IEEE Computer* **37**(7) (2004) 56–64
2. Ofria, C., Wilke, C.O.: Avida: A software platform for research in computational evolutionary biology. *Journal of Artificial Life* **10** (March 2004) 191–229
3. Lenski, R.E., Ofria, C., Pennock, R.T., Adami, C.: The evolutionary origin of complex features. *Nature* **423** (2003) 139–144
4. Adami, C., Ofria, C., Collier, T.C.: Evolution of biological complexity. *Proceedings of the National Academy of Sciences* **97** (2000) 4463–4468
5. Lenski, R.E., Ofria, C., Collier, T.C., Adami, C.: Genome complexity, robustness, and genetic interactions in digital organisms. *Nature* **400** (1999) 661–664
6. Goings, S., Clune, J., Ofria, C., Pennock, R.T.: Kin selection: The rise and fall of kin-cheaters. In: *Proceedings of the International Conference on Artificial Life (ALIFE)*, Boston, MA, USA (September 2004) 303–308
7. Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T.H., Baldassarre, G., Nolfi, S., Deneubourg, J.L., Mondada, F., Floreano, D., Gambardella, L.M.: Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots* **17**(2–3) (2004) 223–245
8. Babaoglu, O., Canright, G., Deutsch, A., Caro, G.A.D., Ducatelle, F., Gambardella, L.M., Ganguly, N., Jelasity, M., Montemanni, R., Montresor, A., Urnes, T.: Design patterns from biology for distributed computing. *ACM Transactions on Autonomous and Adaptive Systems* **1**(1) (2006) 26–66
9. Baronchelli, A., Dall’Asta, L., Barrat, A., Loreto, V.: Strategies for fast convergence in semiotic dynamics. In: *Proceedings of the International Conference on Artificial Life (ALIFE)*. (2006) 480–485
10. Marocco, D., Nolfi, S.: Self-organization of communication in evolving robots. In: *Proceedings of the International Conference on Artificial Life (ALIFE)*. (2006) 178–184
11. Baranski, B., et al.: High-order punishment and the evolution of cooperation. In: *Proceedings of the International Genetic and Evolutionary Computation Conference (GECCO)*, Seattle, Washington, USA (2006) 379–380
12. McPartland, M., Nolfi, S., Abbass, H.A.: Emergence of communication in competitive multi-agent systems: a pareto multi-objective approach. In: *Proceedings of the International Genetic and Evolutionary Computation Conference (GECCO)*, Washington DC, USA (2005) 51–58
13. Eiben, A.E., Schut, M.C., Toma, T.: Comparing multicast and newscast communication in evolving agent societies. In: *Proceedings of the International Genetic and Evolutionary Computation Conference (GECCO)*, Washington DC, USA (2005) 75–81
14. Marocco, D., Cangelosi, A., Nolfi, S.: The role of social and cognitive factors in the emergence of communication: Experiments in evolutionary robotics. In: *Philosophical Transactions of the Royal Society London – A*. Volume 361. (2003) 2397–2421

15. Klyubin, A.S., Polani, D., Nehaniv, C.L.: Tracking information flow through the environment: Simple cases of stigmergy. In: Proceedings of the International Conference on Artificial Life (ALIFE). (2004) 563–568
16. Lungarella, M., Sporns, O.: Mapping information flow in sensorimotor networks. *PLoS: Computational Biology* **2(10)**: e144 (2006)
17. Smith, R.G., Davis, R.: Frameworks for cooperation in distributed problem solving. In: Distributed Artificial Intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1988) 61–70
18. Lynch, N.A.: Distributed Algorithms. Morgan Kaufmann (1997)
19. Zhao, J., Govindan, R., Estrin, D.: Computing aggregates for monitoring wireless sensor networks. In: Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications. (May 2003) 139–148
20. Knoester, D.B., McKinley, P.K., Beckmann, B., Ofria, C.: Evolution of leader election in populations of self-replicating digital organisms. Technical Report MSU-CSE-06-35, Computer Science and Engineering, Michigan State University, East Lansing, Michigan (December 2006)
21. Knoester, D.B., McKinley, P.K., Ofria, C.: Using group selection to evolve leadership in populations of self-replicating digital organisms. In: Proceedings of the International Genetic and Evolutionary Computation Conference (GECCO), London, UK (July 2007) (to appear).