Cooperative Network Construction Using Digital Germlines

David B. Knoester, Philip K. McKinley, and Charles Ofria
Department of Computer Science and Engineering
Michigan State University
East Lansing, Michigan 48824
{dk,mckinley,ofria}@cse.msu.edu

ABSTRACT

This paper describes a study in the evolution of cooperative behavior, specifically the construction of communication networks, through digital evolution and multilevel selection. In digital evolution, a population of self-replicating computer programs exists in a user-defined computational environment and is subject to instruction-level mutations and natural selection. Multilevel selection links the survival of the individual to the survival of its group, thus encouraging cooperation. The results of experiments using the AVIDA digital evolution platform demonstrate that populations of digital organisms are capable of constructing communication networks, and that these networks can exhibit desired properties depending on the selective pressures used. We also show that the use of a digital germline can significantly improve evolvability of cooperation.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods, and Search*; D.1.3 [**Programming Techniques**]: Concurrent Programming—*Distributed programming*; F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*Self-modifying machines*

General Terms

Experimentation.

Keywords

Digital evolution, cooperative behavior, natural selection, multilevel selection, mutation, germline, biologically-inspired computing.

1. INTRODUCTION

The natural world is replete with examples of organisms of varying complexity cooperating to build complex structures. For example, biofilms are complex extracellular structures that are formed by nearly all species of microorganism [4], while the eusocial *Hymenoptera* (sawflies, wasps, bees, and ants) build complex nests that house many thousands of individuals either above or below ground [11].

Whereas natural organisms have evolved to cooperate in building physical structures, here we use *digital evolution* [17] to evolve digital organisms that cooperatively build a structure of their own, specifically a communication network. In digital evolution, a population of digital organisms exists in a user-defined computational environment. These organisms self-replicate, compete for resources, and are subject to instruction-level mutations and natural selection. Over generations, they can evolve to survive, and even thrive, under extremely dynamic and adverse conditions. In digital evolution, whether a given organism self-replicates and transfers its genetic material to the next generation depends on its environment and its interaction with other organisms. In this study we use AVIDA [18], a digital evolution platform previously used to study the evolution of biocomplexity [1, 13] and phenotypic plasticity [3]. AVIDA is being applied to an increasingly diverse range of problems, from the generation of software behavioral models to the design of distributed systems [10, 14].

To evolve network building, we divide the population into identically-sized subpopulations, or *demes*. Within each deme, organisms are treated as nodes in a network, and are capable of establishing links to other organisms within their deme. We then apply multilevel selection [8] to all demes in the population, periodically evaluating the network constructed by each deme. The replication rate of the deme is dependent on the properties of the network it constructs. The specific network properties selected for, such as link usage and characteristic path length, are user-defined. In the course of our investigations, we discovered that the use of a *digital germline* significantly improves the evolvability of cooperation. In animals, the germline is the sequence of germ cells used to transfer genetic material from parent to offspring [7]. Similar to the germ-soma specialization common in multicellular organisms, a germline in AVIDA provides a single common genetic ancestry for all organisms within a deme. This has the side-effect of homogenizing the inhabitants of each deme, a technique that has been shown effective in evolutionary robotics [22].

The contributions of this work are two-fold. First, we show that the use of multilevel selection alone is not sufficient to evolve network construction, requiring the addition of a germline to provide a common ancestry for cooperating organisms. Second, we show that AVIDA is capable of evolving digital organisms that construct networks, and that the networks thus created can meet user-defined criteria, including the minimization of link usage, diameter, characteristic path length, and clustering coefficient. These results can

be applied to the construction and maintenance of communication networks in distributed systems, including many wireless and overlay network applications. Our long-term goal is to use digital evolution in the design of robust distributed systems that remain effective even under extremely harsh conditions. While evolved solutions may share some of the inherent imperfections of natural organisms, they might also be resilient to unexpected conditions, where human-designed algorithms are limited and/or brittle.

2. RELATED WORK

From large-scale permanent networks such as the Internet, to smaller temporary networks such as ad hoc wireless networks, network creation and topology maintenance algorithms are a fundamental aspect of distributed computing systems. Network creation in distributed systems is often addressed through the use of *overlay networks*, where a communication network is created and maintained by application software. Overlay networks involve only the modification of application software at hosts, without requiring changes to network infrastructure. Approaches to building overlay networks range from distributed hash-tables and unstructured peer-to-peer networks, to bio-inspired gossip-based approaches such as T-Man [12]. Cooperative models of the iterated prisoner's dilemma have also been applied to network creation and topology maintenance [19], and other game-theoretic approaches have demonstrated the emergence of leadership and cooperation in social and economic networks [25]. In contrast to these approaches, where system behavior is specified by humans, our work uses digital evolution to automatically discover cooperative behaviors for network creation. Common to all implementations of network creation in distributed systems is that the agents involved engage in some degree of cooperation, whether explicit (in the case of a human-designed deterministic algorithm), or implicit (in the case of a gossip-based protocol).

Examples of the evolution of cooperative behavior are pervasive in biological systems, and include cooperation among microorganisms, insects, and even in the schooling of fish to avoid predators. Many organisms cooperate to build complex structures, from biofilms [4] to nests [11]. Researchers have focused on understanding the basis for the evolution of cooperation [2,16], with additional insight into how cooperation might arise uncovered through both digital and robotics experiments [6, 21, 22]. Many approaches to the evolution of cooperative behavior in non-biological systems involve *multilevel selection*, the theory that selection acts not only on the individual directly, but also indirectly through the groups that it is a member of [8]. Researchers in evolutionary computation have used various forms of multilevel selection to accurately diagnose malignancy in cancer [24] and improve runtime performance of multi-objective evolutionary algorithms [9]. Additionally, genetic programming has been used to design communications protocols for autonomic systems, where individuals in the population were different protocols [23]. Recently, investigations have focused on the evolution of single-celled to multicellular life [15]. This transition is the point at which competition between individual cells gives way to competition between groups of cells, thus enabling complex multicellular organisms. In [22], Floreano *et al.* evolved cooperative communication through the use of genetically homogeneous groups, in much the same way that biological systems use a germline for reproduction [7]. In this paper, we use an explicit mechanism for digital germlines to evolve cooperative network construction. Germlines are discussed further in Section 4.

3. AVIDA BACKGROUND

Digital Organisms. 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. Within this environment, organisms asynchronously execute the instructions in their genomes. The particular instructions that are executed determine the organism's behavior, including the ability to sense and change properties of their environment.
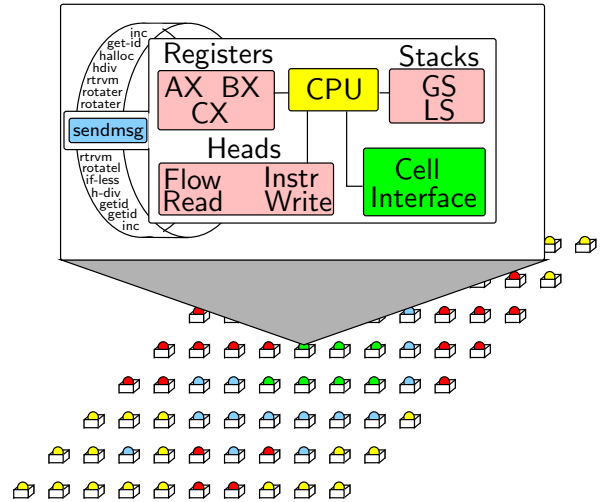
Figure 1: An AVIDA population containing multiple genomes (bottom), and the structure of an individual organism (top).

Instructions within an organism's genome are similar in appearance and functionality to traditional assembly language instructions. These instructions enable an organism to perform simple mathematical operations, such as addition, multiplication, and bit-shifts, as well as to replicate their genome and interact with the organism's environment, for example, by creating a link to a neighboring organism, or outputting a number to the environment. A key property of AVIDA's instruction set that differs from traditional computer languages, however, is that it is not possible to construct a *syntactically incorrect* genome in AVIDA; that is, all possible genomes are "runnable." Hence, while random mutations will produce many genomes that do not perform any meaningful computation, their instruction sequences will still be valid.

During its *replication cycle*, an organism copies its genome to its offspring. Figure 2 depicts the instructions comprising the replication cycle of the default AVIDA organism. As the organism copies its genome to its offspring, mutations may be introduced according to predefined probabilities. 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).
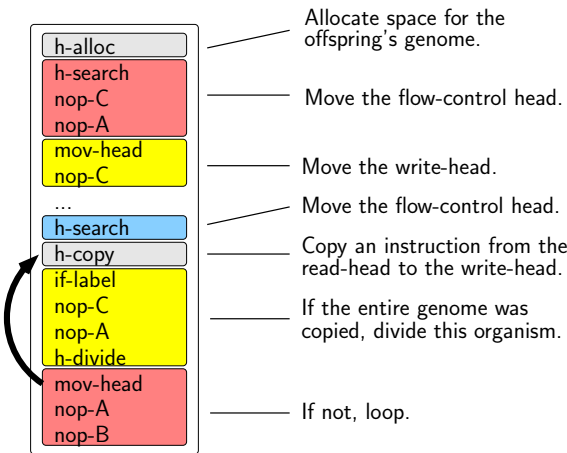
Figure 2: Instruction sequence for the replication cycle of a typical AVIDA organism.

Environment. In Figure 1, we see that each organism in AVIDA lives in a *cell* located in a fixed location within a spatial environment. Each cell can contain at most one organism; organisms cannot live outside of cells. The geometry of the environment defines the neighborhood of each cell, and is user-defined. For example, the environment geometry may be configured as a spatial grid, a torus, or as a well-mixed environment, where all cells are neighbors of each other. Each organism in the environment has a *facing* that defines its orientation. This facing may be used in a number of different ways. For example, an organism can create a network link in the faced direction. The organism can also sense and manipulate its facing via the `get-facing` and `rotate` instructions, respectively. 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 that neighbor 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.

During an AVIDA experiment, the *merit* of a given digital organism determines how many instructions its virtual CPU is allowed to execute relative to the other organisms in the population. A higher merit results in an organism that replicates more frequently, spreading throughout the population. Merit of a digital organism is updated based upon the *tasks* that are performed by the organism. Tasks are designed by the user and reward desirable behavior (they may also punish undesirable behavior), thereby driving natural selection. For example, in order to encourage the creation of a network, a user might define a task that rewards an organism by doubling its merit when it constructs a link to a neighboring organism. Tasks are generally defined in terms of externally visible behaviors of the organisms (their phenotype), rather than in terms of the specific instructions that must be executed by the digital organism's CPU. This approach allows 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 – robustness to mutation, for example.

4. MULTILEVEL SELECTION IN AVIDA

Multilevel selection posits that the survival of the individual is linked to the survival of the group [8]. There are many different ways that these groups may be defined. For example, a group may be defined by a common trait (a trait-group), shared ancestry (clade selection), membership in the same species (species selection), or the interactions between related individuals (kin selection). Multilevel selection has been used in biology to explain the evolution of altruism, where an individual will sacrifice fitness for the benefit of the group, and the evolution of social behavior, particularly in populations of social insects, such as ant and bee colonies [20].

There are two components to multilevel selection within AVIDA. First, the population of digital organisms is divided into distinct subpopulations, called *demes*. Second, we provide a means for these demes to compete with each other based on user-defined criteria. Figure 3 depicts an environment that has been subdivided into sixteen demes. As with the organisms within each deme, entire demes replicate and compete for resources. When a deme replicates, another deme from the population is selected as its target. The organisms in the target deme are removed, and a subset of the organisms from the source are cloned and placed into the target. We call this procedure ReplicateDemes. In Figure 3, we show two demes replicating - Organisms within the target demes will be removed from the population, and the target demes will be repopulated from their respective source demes. Such replication can be triggered by a deme's behavior. In this study, for example, deme replication may be triggered when its constituents have constructed a connected network. Deme replication may also be combined with a deme-level merit, where all individuals within the deme receive additional merit based on the behavior of the deme as a whole. Similar to the individual merit described in Section 3, a deme merit increases the number of virtual CPU instructions its inhabitants are allowed to execute, relative to other demes in the population. Use of a deme merit thus provides a selective pressure that operates on entire demes. As will be shown in Section 6, deme merit is an effective method for guiding the evolution of group behavior.
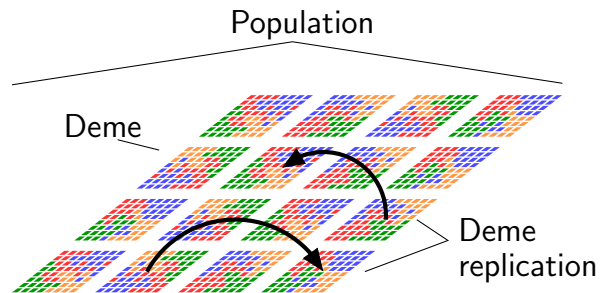
Figure 3: Depiction of an AVIDA population of sixteen demes. Demes are isolated subpopulations, each capable of replication. When a deme replicates, it replaces a randomly selected target deme.

For the study described in this paper, we extended ReplicateDemes to use a germline; we call this new procedure GermlineReplication. In animals, genetic material is transferred from parent to offspring along the germline [7]. *Germ cells* are distinct from *somatic cells*, which form the body of

an organism. Mutations to an organism's germline can be passed on to its offspring, while mutations to an organism's somatic cells will typically only affect its host. Figure 4 illustrates GermlineReplication. Beginning from an ancestral germline g_0, the "parent" deme is seeded with an organism generated from the latest germ. During the course of the experiment, the somatic organisms within this deme replicate, compete for resources, and may experience mutations. Once a deme replication is triggered, all organisms within the parent deme are killed, and the parent is re-seeded from its germline, g_0. The latest germ from g_0 is then mutated, producing a new germline, g_1. An "offspring" deme is randomly selected from the population, any organisms currently living in this deme are killed, and the new germ from g_1 is used to seed the offspring deme. As with ReplicateDemes, a deme-level merit may also be used with ReplicateDemes. In this case, the merit is associated with the germline.

(1) Parent deme seeded with germline g_0

(2) Replication triggered, parent reseeded from g_0

$g_0 \rightarrow g_1$ (3) g_0 mutated, produces g_1; offspring seeded from g_1
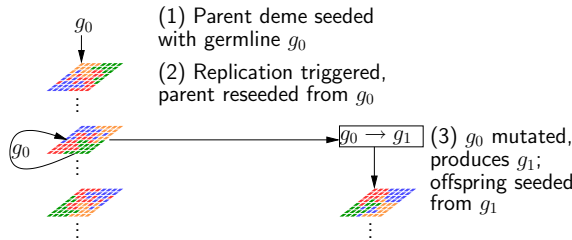
Figure 4: Example of GermlineReplication.

5. EXPERIMENTAL PROCEDURES

Extensions to AVIDA. To enable digital organisms within AVIDA to build a network, we implemented five new virtual CPU instructions. Each of these instructions can be inserted into a genome through mutation during deme replication. These instructions, summarized in Table 1, enable organisms to build links to neighboring organisms, sense their current links, perform conditional logic based on their links, and determine their location within their environment. The current implementations of these instructions do not support parallel links (multiple links between the same two organisms), links do not expire, and there is no cost associated with creating a link; we plan to remove these limitations in future work. It may be useful to think of the network and its links through an analogy to UNIX sockets: organisms in demes are hosts that are able to establish socket connections to their neighbors to support communication.

A typical network creation process for a 4×4 deme is depicted in Figure 5. Here we see the network at different stages throughout the deme's lifetime. In Figure 5(a), the deme was seeded with a single organism, which subsequently replicated twice and created a link to one of its offspring. In Figure 5(b), the original organism and its offspring continue to replicate, and more links are built. Finally, in Figure 5(c), the deme is filled to capacity and all organisms are connected in a single network. At this point, if a replication trigger had been defined to replicate a deme when the deme was full and the network was connected, the deme would be replicated.

Experimental Setup. Except for the different deme replication triggers, the experiments presented here were configured identically; relevant configuration options are summarized in Table 2. Values were experimentally se-

Table 1: Instructions added to AVIDA to support network construction.

Instruction	Description
CR-LINK	Create a link between the caller and the currently-faced neighbor.
RD-LINKS	Retrieve the IDs of all the organisms linked to the caller.
GET-LPOS	Retrieve the location of the caller within the environment.
IF-LINKED[a]	Test if there is a link in the direction currently faced.
IF-NOT-LINKED	Test if there is not a link in the direction currently faced.

[a]Conditionals (IF-LINKED, IF-NOT-LINKED) execute the subsequent instruction in the caller's genome if they evaluate to true; otherwise, the subsequent instruction is skipped.
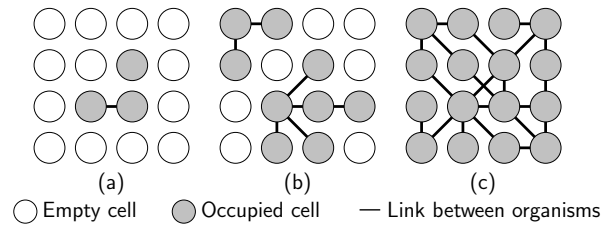

○ Empty cell   ● Occupied cell   — Link between organisms

Figure 5: Sample networks constructed by the organisms within a deme.

lected to balance computation time and repeatability, except for the copy mutation rate, which was unchanged from the AVIDA default. For each experiment presented, 20 distinct AVIDA trials were conducted.

Table 2: AVIDA configurations used for the experiments described in this study.

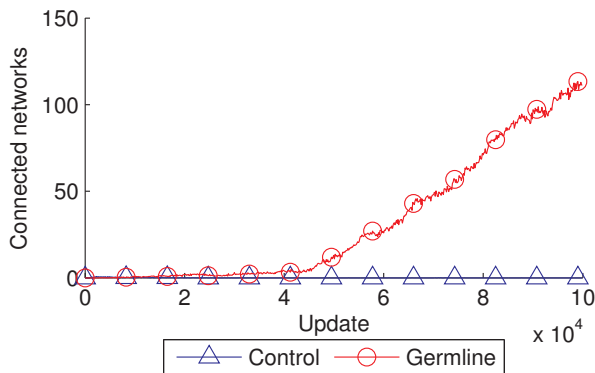| Configuration	Value
Trials per experiment	20
Max. population size	19,600
Number of demes	400
Max. deme size	49
Max. deme age	150 updates
Environment geometry	7×7 spatial grid
Copy mutation rate	0.0075

We note that when using GermlineReplication, mutation only occurs during deme replication. However, replication is not triggered until the deme performs a specified behavior. During the early stages of an AVIDA trial this presents a "bootstrapping" problem, as the default ancestral organism only contains the instructions necessary for self-replication. To bypass this problem, we include an additional replication trigger that unconditionally replicates each deme based on its age, measured in updates. An *update* is the standard unit of time in AVIDA, and corresponds to the time required for each organism to execute, on average, 30 instructions. The age of a deme is measured from the current update to the update at which the deme was seeded or replicated, whichever is most recent.

6. RESULTS

In this section we present the results of our experiments, progressing from relatively simple single-criteria problems to more complex multiobjective optimization problems. We first focus on the evolution of organisms that construct connected networks, where there is at least one path between every pair of nodes. Next, we show that simple selective pressures can alter characteristics of these networks, including link count, diameter, characteristic path length, and clustering coefficient. We then show that more complex selective pressures can be used to build networks that balance multiple criteria, for example, building networks that have both a low link count and a small diameter.

Evolution of Connected Networks. Our first series of experiments focused on the evolution of organisms that build a connected network, that is, one in which there exists a path between all pairs of organisms. For these experiments, we implemented a replication trigger that would replicate any deme whose inhabitants had constructed a connected network.

Figure 6 plots the mean (across all 20 trials) number of connected networks constructed under two different configurations. The first, a control experiment, used ReplicateDemes, while the second experimental treatment used GermlineReplication. As shown, very few connected networks were constructed under the control, and this behavior never reached fixation within any population. However, when using GermlineReplication, all 20 trials were able to regularly construct connected networks. We note that under GermlineReplication, the number of connected networks constructed at each time step continued to increase with time, indicating that the population was becoming more efficient at constructing networks. This result indicates that using a germline significantly increases the evolvability of cooperation. As such, all remaining experiments use the GermlineReplication competition strategy, configured to replicate a deme as soon its network is connected.

Figure 6: Mean number of connected networks constructed under ReplicateDemes (control) and GermlineReplication (germline) configurations, per 100 updates. GermlineReplication was significantly more effective at evolving network construction.

Reducing Link Usage. In the previous experiment, while the AVIDA populations were successful at constructing connected networks, these networks contained on average 114 links – over twice as many links as what are needed to construct a connected network of 49 organisms. In this experiment we investigated the evolution of digital organisms to construct networks with reduced link usage. To encourage the construction of networks with fewer links, we augmented replicated demes with a merit that increased with fewer links. In this and subsequent experiments, AVIDA was extended with algorithms to calculate merit based on group behavior. Specifically, merit was set to the following:

$$Merit = \left(E_{max} - |E| + 1 \right)^2,$$

where E_{max} is the maximum number of links that could be in the network (based on our knowledge of the underlying environmental geometry and the size of each deme) and $|E|$ is the number of links present in the network (the additional $+1$ is used to differentiate between demes that replicate due to age, and those that replicate due to constructing a connected network with the maximal number of edges).
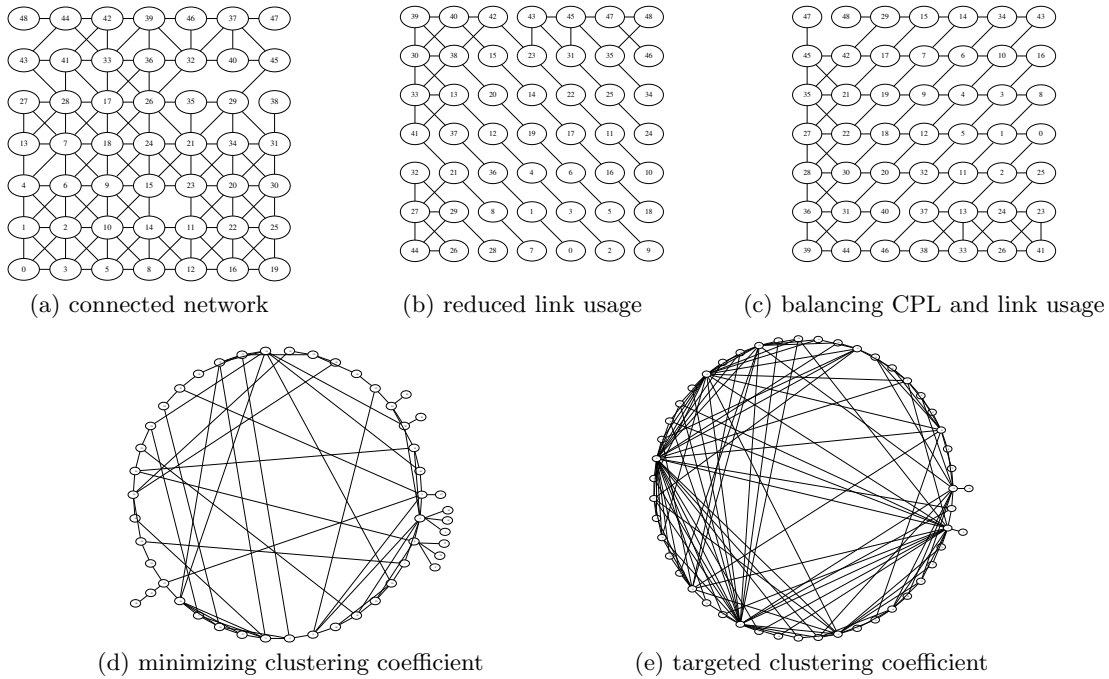
Figure 7 plots the mean number of links used in constructed networks, and the mean number of connected networks actually constructed. Here we see that these trials quickly evolved to use a large number of links (an effective strategy for building a connected network), and further evolution reduced the number of links that were used. At the end of this experiment, constructed networks contained an average of 86 links (1.75 links per organism), while the best performing single AVIDA trial averaged 62.1 links per network (1.2 links per organism). As before, the number of networks being constructed continues to increase with time, indicating increasing efficiency. Figure 8(b) depicts a sample network constructed by the dominant genotype from this experiment, while Figure 8(a) depicts a sample network constructed from our initial experiment where reduced link usage was not rewarded.
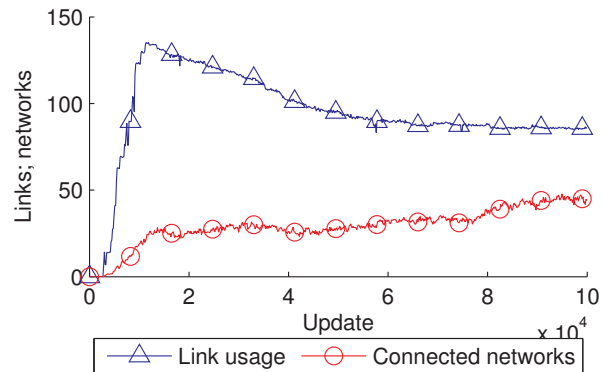
Figure 7: Mean number of links used to construct networks, per 100 updates. Selecting for networks with fewer links results in organisms that use links sparingly, while still building a connected network.

In examining the genomes responsible for the construction of these networks, we uncovered three primary strategies by which networks with different properties were created. First, organisms would sense their location within the environment, conditionally linking to neighboring organisms based on their location. For example, in Figure 8(b) we see conditional behavior based on whether the organism is along the west or north edge of the deme. Second, organisms would frequently use the ROTATE instructions (described in

(a) connected network   (b) reduced link usage   (c) balancing CPL and link usage

(d) minimizing clustering coefficient   (e) targeted clustering coefficient

Figure 8: Example networks constructed by evolved organisms.

Section 3) to unconditionally face a given direction, which when combined with location-awareness facilitates network creation. The third strategy is more stochastic in nature, where organisms would create links only if they were not already linked to at least one neighbor. Although this is an effective strategy for creating a connected network, it is difficult to construct networks with desired properties in this way. We suspect that other features of the organism's lifecycle, such as gestation time, frequency of attempting link creation, and possibly communication with neighboring organisms are influencing their behavior. Further investigation is needed to verify the specific strategies being used.

Reducing Diameter and Characteristic Path Length. In some situations, dense communication networks are more desirable than sparse networks, as they generally lead to shorter message latencies. In this experiment, we investigate the evolution of digital organisms to construct networks that reduce two properties related to network density: diameter and characteristic path length. Whereas the diameter of a network is the maximum distance between any pair of nodes in the network, the characteristic path length (CPL) of a network is the mean pairwise distance between all nodes in the network; both are measured in terms of the number of links that must be traversed. To encourage reduction of diameter and CPL, we again augmented replicated demes with a merit indirectly proportional to these two measures. Specifically, for diameter-reducing trials merit was set to the following:

$$Merit = \Big(|V| - D(N) + 1\Big)^2,$$

where $|V|$ is the number of organisms present in the network and $D(N)$ is the calculated diameter of the network. Simi-

larly, for CPL-reducing trials, merit was set to the following:

$$Merit = \Big(CPL_{max} - CPL(N) + 1\Big)^2,$$

where $CPL(N)$ is the calculated CPL of the network and CPL_{max} is the maximum possible value of the CPL (given our knowledge of the underlying environmental geometry and size of the demes).

Figure 9 plots the mean diameter and mean CPL for these two experiments. As can be seen here, both diameter and CPL approach their optimal values (for the spatial grid used in these experiments, the optimal diameter is 6, while the optimal CPL is 3.5). In each experiment, the minimization of diameter and CPL come at the expense of link usage, which averaged 138. In an experiment described later, we attempt to balance such competing concerns.

Clustering Coefficient. The *clustering coefficient* of a network is a measure of connectedness. Formally, the clustering coefficient is the ratio of the number of links between nodes in a given node's neighborhood to the number of possible links between those nodes, averaged over all nodes in the network. In a sense, it measures how many neighbors of a given node are themselves neighbors of each other. In this experiment, we investigate the evolution of digital organisms to construct networks that have specified clustering coefficients. To encourage the maximization of the clustering coefficient, merit was set to the following function:

$$Merit = \Big(100.0 \cdot CC(N)\Big)^2,$$

where $CC(N)$ is the calculated clustering coefficient of the constructed network. To minimize the clustering coefficient of connected networks, merit was set to:

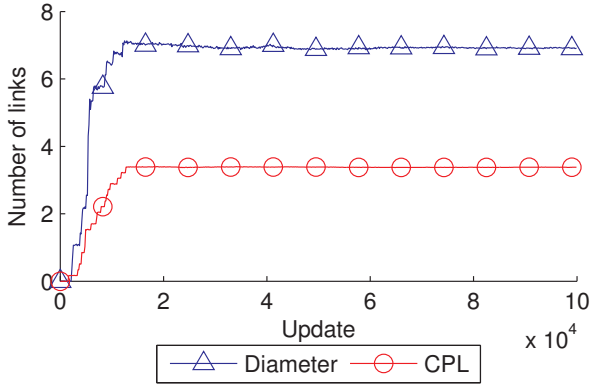$$Merit = \Big(101.0 - 100.0 \cdot CC(N)\Big)^2.$$

Figure 9: Mean number of links for diameter-reducing and characteristic path length (CPL)-reducing experiments, per 100 updates. In both cases, organisms evolved to construct networks that approached their optimal diameter and CPL.
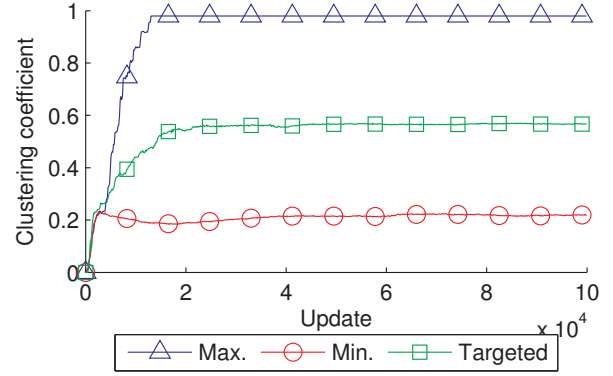

Figure 10: Mean clustering coefficients for three different experiments, per 100 updates. Organisms evolved to construct connected networks with maximal, minimal, and targeted (0.5 desired, 7% error) clustering coefficients.

Finally, we also wished to evolve organisms that constructed networks with an arbitrarily-selected clustering coefficient of 0.5. For this purpose, we set merit to:

$$Merit = 0.01^{2 \cdot |0.5 - CC(N)|}.$$

One difference between this experiment and all others is that the underlying environmental geometry was changed from a spatial grid to a well-mixed one. This change enables any organism to link to any other organism within the same deme, and is necessary to reach higher clustering coefficients.

Figure 10 plots the three different mean clustering coefficients for the minimizing, maximizing, and targeted coefficient treatments. Here we see that each treatment reached its steady-state value within 20,000 updates, with the minimizing treatment reducing the clustering coefficient to 0.2, the maximizing treatment increasing the clustering coefficient to 0.99, and the targeted treatment produced a clustering coefficient of 0.57, an error of approximately 7% from the requested clustering coefficient. Figure 8(d) and Figure 8(e) depict sample networks constructed by the dominant genotypes from the minimizing and targeted experiments.

Balancing CPL and Link Usage. To this point, we have shown that digital evolution is capable of evolving cooperating organisms that construct networks while optimizing for a single criteria. Next, we present initial results showing the evolution of organisms to construct networks that attempt to balance multiple, possibly conflicting, objectives. Specifically, here we evolve organisms that are rewarded for constructing networks that minimize both their characteristic path length and their link usage. These two objectives are conflicting – A highly connected network (one with a low CPL) is likely to use a large number of links, while a network containing few links is likely to have a high CPL.

Similar to other multiobjective evolutionary optimization (MOEO) problems [5], this problem exhibits a sensitive dependence on the specific selective pressures (fitness functions, merit, etc.) that are used. Here, three different treatments are examined, each using a different formulation for the merit applied to replicated demes. Specifically, in the

following *multiplicative* treatment, merit was set to:

$$Merit = \left((CPL_{max} - CPL(N) + 1) \times (E_{max} - |E| + 1) \right)^2.$$

The *additive* treatment is similar, changing only the internal multiplication of the CPL and link count from multiplication to addition. Finally, the *normalized* treatment extends the additive model by normalizing the CPL- and link-based terms to the range 0 to 100:

$$Merit = \left(100 \cdot \frac{CPL_{max} - CPL(N)}{CPL_{max}} + 100 \cdot \frac{E_{max} - |E|}{E_{max}} + 1 \right)^2.$$

Figure 11 plots the mean CPL and link count for each of these three different fitness functions. Here we see that the normalized merit treatment reached its steady-state value in 10,000 updates, while the multiplicative and additive treatments continued to minimize their link usage, at the cost of an increased CPL. The different behaviors that result from these fairly minor changes to fitness functions are typical of MOEO problems, though further analysis is required to compare these results to the Pareto-optimal set. Figure 8(c) is a sample network constructed by organisms attempting to balance CPL and link usage. As can be seen in this network, they made extensive use of diagonals (which are advantageous when using the grid environmental geometry), but also included horizontal edges to further reduce their CPL.

7. CONCLUSION

We have demonstrated that digital evolution, in combination with multilevel selection and germlines, can produce a relatively complex cooperative behavior, specifically network construction. Furthermore, we have shown that this behavior can be produced by selecting for properties of the network, rather than by specifying individual-level criteria. This work is a first step towards synthesizing topology maintenance algorithms for distributed systems, and hints that evolutionary computation, specifically multiagent approaches such as digital evolution, has promise in the design of future distributed computing systems.

(a)

(b)
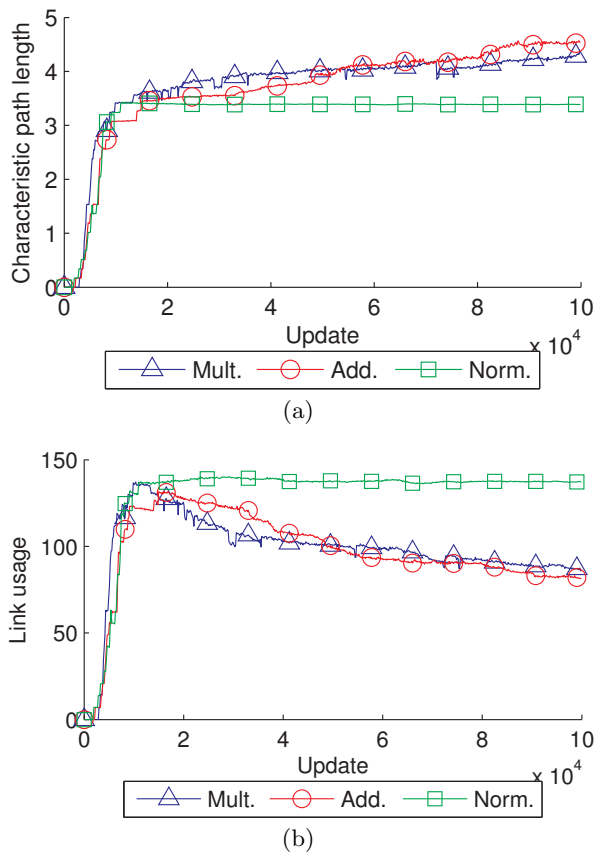
Figure 11: Mean characteristic path length (CPL) (a) and link usage (b) under three different experiments that balance CPL and link usage, per 100 updates.

8. REFERENCES

[1] C. Adami, C. Ofria, and T. C. Collier. Evolution of biological complexity. *Proc Natl Acad Sci USA*, 97:4463–4468, 2000.

[2] R. M. Axelrod. *The Evolution of Cooperation*. Basic Books, 1984.

[3] J. Clune, C. Ofria, and R. T. Pennock. Investigating the emergence of phenotypic plasticity in evolving digital organisms. In *European Conference on Artificial Life (ECAL)*, pages 74–83, 2007.

[4] M. E. Davey and G. A. O'Toole. Microbial biofilms: from ecology to molecular genetics. *Microbiology and Molecular Biology Reviews*, 64(4):847–867, 2000.

[5] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, 2001.

[6] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, and L. M. Gambardella. Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2–3):223–245, 2004.

[7] E. Eddy. The germ line and development. *Developmental Genetics*, 19:287–289, 1996.

[8] D. S. Wilson. Altruism and organism: Disentangling the themes of multilevel selection theory. *The American Naturalist*, 150:S122–S134, 1997.

[9] O. Giel and P. K. Lehre. On the effect of populations in evolutionary multi-objective optimization. In *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*, pages 651–658, 2006.

[10] H. J. Goldsby, B. H. C. Cheng, P. K. McKinley, D. B. Knoester, and C. Ofria. Digital evolution of behavioral models for autonomic systems. In *Proceedings of the International Conference on Autonomic Computing (ICAC)*, 2008.

[11] D. Grimaldi and M. S. Engel. *Evolution of the Insects*. Cambridge University Press, 2005.

[12] M. Jelasity and O. Babaoglu. T-man: Gossip-based overlay topology management. In *Proceedings of the Workshop on Engineering Self-Organising Applications (ESOA)*, 2005.

[13] R. E. Lenski, C. Ofria, R. T. Pennock, and C. Adami. The evolutionary origin of complex features. *Nature*, 423:139–144, 2003.

[14] P. K. McKinley, B. H. C. Cheng, C. Ofria, D. B. Knoester, B. Beckmann, and H. Goldsby. Harnessing digital evolution. *IEEE Computer*, 41(1), 2008.

[15] R. Michod. On the transfer of fitness from the cell to the multicellular organism. *Biology and Philosophy*, 20(5):967–987, 2005.

[16] M. A. Nowak. Five rules for the evolution of cooperation. *Science*, 314(5805):1560–1563, 2006.

[17] C. Ofria and C. Adami. Evolution of genetic organization in digital organisms. In *Proceedings of DIMACS Workshop on Evolution as Computation*, 1999.

[18] C. Ofria and C. O. Wilke. Avida: A software platform for research in computational evolutionary biology. *Journal of Artificial Life*, 10:191–229, 2004.

[19] J. C. Scholz and M. O. W. Greiner. Topology control with IPD network creation games. *New J. Phys.*, 9(185), 2007.

[20] T. D. Seeley. Honey bee colonies are group-level adaptive units. *The American Naturalist*, 150:S22–S41, 1997.

[21] G. Werner and M. Dyer. Evolution of communication in artificial organisms. In *Artificial Life II*, pages 659–687. Addison-Wesley Pub., 1992.

[22] D. Floreano, S. Mitri, S. Magnenat, and L. Keller. Evolutionary conditions for the emergence of communication in robots. *Current Biology*, 17(6):514–519, 2007.

[23] L. Yamamoto and C. F. Tschudin. Experiments on the automatic evolution of protocols using genetic programming. In *Proceedings of the IFIP Workshop on Autonomic Communication (WAC)*, 2005.

[24] E. M. Zechman and S. R. Ranjithan. Multipopulation cooperative coevolutionary programming (MCCP) to enhance design innovation. In *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*, pages 1641–1648, 2005.

[25] M. G. Zimmermann and V. M. Eguiluz. Cooperation, social networks, and the emergence of leadership in a prisoner's dilemma with adaptive local interactions. *Physical Review E*, 72(5), 2005.