Evolution of Division of Labor in Genetically Homogenous Groups*

Heather J. Goldsby, David B. Knoester, and Charles Ofria
Department of Computer Science and Engineering
Michigan State University
East Lansing, Michigan 48824 USA
hjg@msu.edu, dk@msu.edu, ofria@msu.edu

ABSTRACT

Within nature, the success of many organisms, including certain species of insects, mammals, slime molds, and bacteria, is attributed to their performance of division of labor, where individuals specialize on specific roles and cooperate to survive. The evolution of division of labor is challenging to study because of the slow pace of biological evolution and imperfect historical data. In this paper, we use digital evolution to evolve groups of clonal organisms that exhibit division of labor. We then investigate what mechanisms they use to perform division of labor (i.e., location awareness or communication) and discover that it varies according to the type of roles being performed. Lastly, we created an environment where groups of organisms needed to complete a set of tasks, but could do so as either generalists or specialists. We varied the costs of switching tasks and determined that increased costs can result in the evolution of division of labor. Moreover, a group used as a case study exhibited both division of labor and cooperative problem decomposition, where members of the group shared partial solutions to solve the full set of problems. This approach has the potential to inform predictions in biological studies, as well as achieving division of labor when using evolutionary computation to solve more complex engineering problems.

Categories and Subject Descriptors

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

General Terms

Experimentation.

Keywords

Digital evolution. Avida. Eusociality. Division of Labor.

*This work has been supported in part by NSF Grants CCF-0643952, CNS-0751155 and CCF-0820220, and the DARPA FunBio Program.

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.

1. INTRODUCTION

Division of labor, where individuals specialize on specific roles and cooperate to survive, is hailed as a strategy central to the success of many organisms, including certain species of insects [3, 9, 19, 23], mammals [3, 19], slime molds [2, 3, 19], and bacteria [3]. Within nature, eusocial organisms are renowned for exhibiting *reproductive division of labor*, where members of the reproductive caste (i.e., queens) produce offspring and members of the non-reproductive caste care for the brood and perform other duties central to the maintenance of the eusocial colony [9]. Moreover, many eusocial organisms, such as leaf-cutter ants [23], bumblebees [9], and aphids [17], also exhibit *task-related division of labor*, where individuals specialize on performing a particular task. For example, non-reproductive worker bumblebees specialize to perform roles that include foraging, caring for the brood, building honeypots, guarding the hive, or cooling the hive through fanning [9]. In this paper, we use artificial life to explore questions regarding the evolutionary conditions that give rise to task-related division of labor that are challenging, if not impossible, to study using biological approaches.

Extensive research has been done to better understand division of labor (e.g., [2, 3, 9, 19, 23]). Among other results, scientists have classified patterns of division of labor. Some common patterns include *temporal specialization*, where the individual's age predicts which task it specializes on [19]; *spatial specialization*, where the distance from the hive center predicts the task specialization of the worker [9]; and *morphological specialization*, where the phenotype of the worker is correlated with the task it specializes on [23]. Additionally, scientists have identified some of the mechanisms used by organisms to allocate tasks, such as differences in behavioral response thresholds, hormones, and genetic variation [20]. However, exploring division of labor in an evolutionary context remains challenging owing to the slow pace of evolution and imperfect historical data regarding species that have evolved division of labor.

In this paper, we use digital evolution to perform a preliminary study in exploring the evolution of division of labor. Specifically, we use Avida [16], a digital-evolution platform previously used to study topics including the origin of complex features [12] and the evolutionary design of modularity [15]. Within an Avida experiment, a population of self-replicating computer programs exists in a user-defined computational environment and is subject to mutations and natural selection. These "digital organisms" execute their genome to perform tasks that metabolize resources in the environment, interact with neighboring organisms, and selfreplicate. Avida satisfies the three requirements for an evolutionary system (replication, variation (mutation), and differential fitness (competition) [4]), while enabling rapid evolution, unlimited experimental control, and perfect data.

For this study, we explore the evolution of task-related division of labor within genetically homogeneous groups. We create groups of Avida organisms called *demes*, where all organisms within a deme are identical. These homogeneous demes are similar to clonal groups, such as some slime molds [2, 3, 19] and some colonies of aphids [17]. For clonal groups to exhibit division of labor, they must exhibit *phenotypic plasticity*, where organisms with the same genome behave differently. We address three questions related to the evolution of division of labor. First, if we require the organisms to be *specialists* that only perform one task each, then can demes evolve to engage in division of labor? We examine this question in two different environments that vary the number and difficulty of the tasks. Second, is there a relationship between the mechanism used to perform division of labor and the type of environment? Lastly, under what environmental conditions will division of labor "naturally" evolve? In these experiments, we let the organisms each perform as many or few types of tasks as they choose and examine how the demes react to varying the cost of switching between tasks.

This approach can be used to inform biological studies of cooperation, such as those performed by Dornhaus *et al.* for honeybees [5]. Additionally, this technique can serve as a means to achieve division of labor within artificial life in order to solve engineering problems, such as developing multiple software components that must interact to achieve an overall objective [14], as well as cooperation among heterogeneous sensor networks or robots [11].

2. RELATED WORK

Within evolutionary computation, there are two main bodies of work related to the study of division of labor. First, *automatic problem decomposition* aims to divide a problem into simpler subproblems, individually solve those subproblems, and combine their solutions to obtain a solution to the overall problem [18]. Evolving specialists that solve different subproblems can be considered a form of division of labor. Second, numerous different approaches address the evolution of controllers for multiagent systems [1, 7, 22, 24]. These agents may specialize and thus exhibit division of labor. Here, we provide details regarding the most closely related pieces of work in these two areas.

Numerous techniques have been proposed to automate the evolution of problem decomposition [8, 13, 18, 24]. These techniques evolve specialists that produce solutions to subproblems. The overall solution is a group of specialists comprising either one representative member of each species [8, 18, 24] or a group selected by the evolutionary algorithm [13]. The primary approach used for problem decomposition is the cooperative coevolution architecture [18], which evolves two or more species in isolated populations. During fitness evaluation, individuals from one species are tested with representatives from each of the other species. Our approach differs from this prior work in that we are evolving clonal groups in isolated subpopulations that evolve to use phenotypic plasticity to perform division of labor.

Waibel, Keller, and Floreano provide a comprehensive overview of research approaches that evolve controllers for multiagent systems [22]. Their classification highlights whether the group is homogeneous (e.g., [1, 7]) or heterogeneous (e.g., [24]) and whether the level of selection is at the level of the individual (e.g., [7]) or whole group (e.g., [1, 24]). Our approach uses a homogeneous group with group-level selection. Although multiagent approaches do evolve specialists, the majority of approaches differ from our current approach in that their objective focuses exclusively on producing division of labor in order to solve engineering problems. One notable exception is a study of the effect of genetic variability on the evolution of division of labor [21]. This study uses evolutionary computation to enable differing levels of genetic relatedness in a colony and understand how this affects the evolution of division of labor. This is a complementary approach to the one proposed here, since they use heterogeneous groups and study the role of genetic variability, whereas we use homogeneous groups and study the role of task switching costs.

3. METHODS

Digital Organisms. Figure 1 depicts an Avida population and the structure of an individual organism [16]. Each digital organism is defined by a circular list of instructions (its *genome*), a virtual CPU, and its position in a common virtual environment. Within this environment, organisms 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.

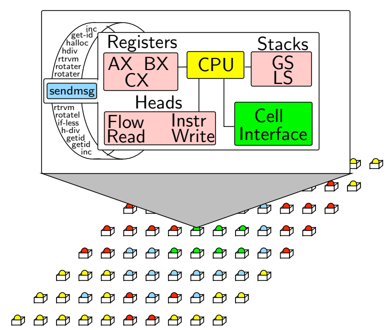

Figure 1: An AVIDA population containing multiple genomes in a spatial environment (bottom), and the structure of an individual organism (top).

The Avida instruction set performs basic computational tasks (addition, multiplication, and bit-shifts), controls execution flow, enables communication, and allows for replication. In this study, the instruction set also included several instructions developed for the evolution of distributed problem solving [14]; these instructions are summarized in Table 1. (Other instructions are described in [16].)

Organisms in Avida can perform *tasks*, which enable them to metabolize resources from their environment to gain additional virtual CPU cycles. For the experiments described in this paper, we created tasks for each possible role. An organism fulfills a role by performing its associated task. For some experiments, we used *role-ids*, a mechanism whereby an organism sets a special-purpose virtual CPU register to an integer value, to indicate the role that the organism performs. For others, we required the organisms to perform

Table 1: Coordination instructions for this study.

Instruction	Description
repro	Enables single-instruction replication.
send-msg	Sends a message to the neighbor currently faced by the caller.
retrieve-msg	Loads the contents of a received message into the caller's virtual CPU.
rotate-left-one	Rotate this organism counter-clockwise one step.
rotate-right-one	Rotate this organism clockwise one step.
get-role-id	Sets register BX to the value of the caller's role-id register.
set-role-id	Sets the caller's role-id register to the value in register BX.
bcast1	Sends a message to all neighboring organisms.
get-cell-xy	Sets register BX and CX to the (x, y) coordinates of the caller.
collect-cell-data	Sets register BX to the value of the cell data where the caller lives.

bitwise Boolean logic operations on 32-bit integers to denote their role. For example, an organism could execute a series of instructions that perform the task for logical AND.

Tasks consume resources. In most Avida experiments, these resources are *unlimited*, such that they are always available. However, resources may also be *limited*, such that the amount consumed is dependent on the amount of the associated resource that is currently available. These limited resources are set up as the computational equivalent of a chemostat, where each resource flows into the environment at a constant rate, while at the same time a small percentage (typically 1%) of the available resource flows out, limiting total accumulation. Metabolizing these resources can affect the *metabolic rate* of organisms, which determines the rate at which its virtual CPU will execute relative to the other organisms in the population. In this study, all organisms in the population have identical metabolic rate, and the performance of tasks, while consuming resources, does not affect the execution speed of individuals.

Demes. To study group behavior with Avida, we first divide the population of digital organisms into distinct subpopulations, called *demes*. Figure 2 depicts an environment that has been subdivided into sixteen demes. As with individual organisms, entire demes replicate and compete for space. 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. In Figure 2, we show two demes replicating; organisms within the target demes are removed from the population, and are replaced by organisms from the source demes.

In this study, we make use of two different forms of deme replication. First, for some experiments, we use *tournament selection* in which demes compete every 100 updates based on a fitness function, where a deme's success is determined by the behavior of its constituent organisms.[1] Each tournament contains a set of demes selected at random, and the deme with greatest fitness is replicated (ties are broken randomly). Second, some of our experiments use *deme replication*, which enables the asynchronous replication of demes. Such replication is triggered by the collective behavior of individuals within a deme. For example, a deme could be set to replicate once its constituents have consumed a certain amount of resource.

[1] An *update* is the unit of experimental time in Avida corresponding to approximately 30 instructions per organism.

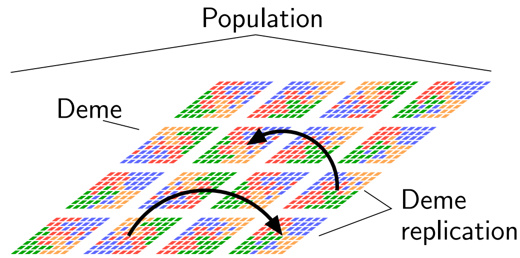

Figure 2: Depiction of an AVIDA population of sixteen demes. Demes are isolated subpopulations, each capable of replication. When a deme replicates, it replaces a selected target deme.

Germlines. Because our study examines the evolution of homogeneous groups, we attached a *germline*, or heritable lineage, to each deme in the Avida population [10]. In most multi-cellular organisms, genetic material is transferred from parent to offspring along the germline [6]. *Germ cells* are distinct from *somatic cells*, which form the body of an organism. Mutations to an organism's germline are passed on to its offspring, while mutations to an organism's somatic cells will typically only affect its host. Figure 3 illustrates this process. 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 and compete for resources. 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. Next, an "offspring" deme is selected from the population (either the 'loser' of the tournament or randomly for deme replication), any organisms currently living in this deme are killed, and the new germ from g_1 is used to seed the offspring deme.

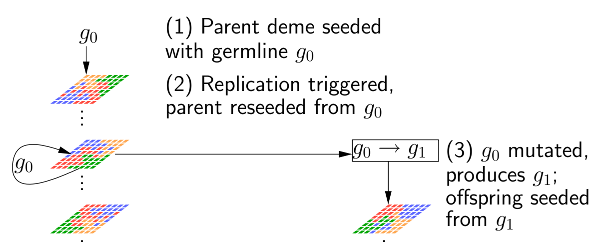

Figure 3: Depiction of the deme replication process using germlines.

4. EXPERIMENTAL RESULTS

For each experiment, we conducted 30 trials to account for the stochastic nature of evolution. Each Avida population comprised 400 demes. Each deme was a 5x5 toroidal grid that could contain up to 25 clonal organisms for an overall population limit of 10,000 individuals. All genomes were fixed at a length of 100 instructions (no insertion or deletion mutations). Mutations occurred to the germline when the deme replicated and the mutation rate was set at 1.0 genomic. When used, tournaments were of size 5.

4.1 Can Avida evolve demes that perform division of labor?

Our first two experiments test whether clonal demes are able to evolve strategies for performing division of labor. Because each organism in the deme has the same genome, successful deme strategies must include both the instructions to perform all of the tasks and also the instructions to coordinate their division of labor. For these experiments, we varied both the number and complexity of the roles. In all cases, demes are rewarded based on how well they exhibit division of labor, which we measure as the number of different tasks done by the organisms at the end of the competition period (every 100 updates).

25 Role Environment. We used the first environment to assess if organisms in demes were able to evolve an extreme form of division of labor, where each organism was required to perform a distinct role. In this experiment, we considered an organism to be performing a given role if it sets its role-id register to a desired value using the set-role-id instruction. The target role-ids are 1 through 25. Because an organism has only one role-id register, it must specialize on just one role at a time. Selection operates based on the diversity of role-ids present during the deme competition period, creating an explicit pressure for division of labor. Specifically, the deme fitness function we used here is:

$$F = \begin{cases} 1+n & \text{if } n < 25 \\ 1+n+r & \text{if } n = 25 \end{cases} \quad (1)$$

where F is the fitness of a deme, n is the number of organisms that have set a role-id, and r is the number of *unique* rewarded role-ids set by organisms in the deme. The first part of the fitness function is designed to reward organisms for selecting any role; whereas, the second part of the fitness function rewards for unique roles within the desired range. Ideally, each organism within a deme will perform a different role and thus the deme will perform all 25 roles.

Figure 4 depicts the maximum (blue line with circles) and mean (red line with triangles) number of tasks performed by the demes averaged across 30 runs. On average, the mean performance was approximately 14.86 (standard error ± 0.67) tasks. However, the organisms within the best demes specialized to perform 23.63 (se± 0.33) different tasks. These results indicate that demes of organisms are able to evolve effective strategies for both specializing to perform roles and also coordinating the distribution of roles.

Logic Environment. Our second set of experiments assesses whether organisms in demes are able to evolve division of labor if the roles require more complex computation. For these experiments, we used a variation of the standard AVIDA logic environment. Each role was a task that required the organism to perform a bitwise logic operation on 32-bit integers. In this case, we used the nine logic operations found in the default AVIDA environment [16]: NOT, NAND, AND, ORN, OR, ANDN, NOR, XOR, and EQU. We made these tasks mutually exclusive – once an organism performed one task, it would no longer receive credit for performing others. Differing from our last environment, these

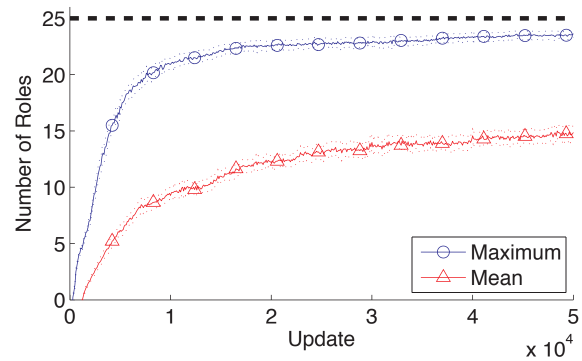

Figure 4: The mean and maximum number of roles performed by demes within the 25 role environment across 30 trials. Notably, some demes approach a perfect score (denoted by a dashed line) of 25.

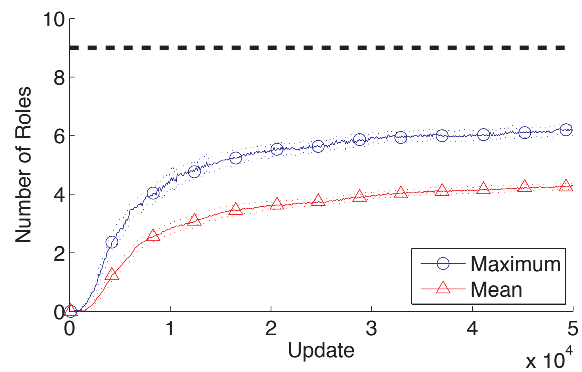

Figure 5: The mean and maximum number of roles performed by demes in the logic environment, where there are nine tasks of increasing complexity.

tasks are more challenging to evolve, but also require less coordination, since there are nine tasks and 25 organisms in a deme. The deme fitness function we used here was:

$$F = 1 + r \quad (2)$$

where F is the fitness of a deme, and r is the number of *unique* tasks performed by the organisms within the deme at the end of the competition period.

Figure 5 depicts the maximum and mean number of tasks performed by the demes averaged across 30 runs. On average, the best deme performed 6.17 (se± 0.16) tasks and the average deme performed 4.27 (se± 0.10) tasks. These values indicate that the problem is challenging, but that the demes are still able to evolve to perform division of labor with complex tasks. We note that when the deme competition pressure is removed, no tasks were performed in any deme, indicating that genetic drift alone is not sufficient to produce division of labor (data not shown).

4.2 What mechanisms are used to perform division of labor?

Given that genetically homogeneous demes of organisms were able to evolve to perform division of labor, we next investigated the mechanisms they used to assign roles and whether these mechanisms change based on environmental context. For the original experiments, we provided the organisms with two different types of mechanisms that we expected them to use to differentiate: message-based communication and location information. Specifically, we provided instructions that, if evolved into a genome and executed by an organism, enabled the organism to send messages to the organism it was facing (send-msg) or broadcast a message to all neighboring organisms (bcast1). For location sensing, we provided organisms with instructions that enabled them to access their x and y coordinates within the deme (get-cell-xy) and to access cell data, a random 32-bit integer associated with their location (collect-cell-data).

To better understand how the mechanisms were used to perform division of labor in different environments, we performed additional experimental treatments that removed one or more coordination capabilities. For each environment, we performed three additional treatments: (1) The organisms had location information, but no messaging capabilities (location only). (2) The organisms had messaging capabilities, but no location information (messaging only). (3) The organisms were deprived of messaging capabilities and location information (none). Comparing the performance of these three treatments to the original treatment that included both messaging capabilities and location information (location & messaging) illuminates which mechanisms are key to the success of the organisms.

25 Role Environment. First, we examined how the organisms performed division of labor in the 25 role environment. Figure 6 depicts the mean (A) and maximum (B) performance of the demes for the coordination mechanism treatments averaged across 30 trials. In general, the location only treatment performs almost identically to the original location & messaging treatment. Likewise, the messaging only treatment performs comparably to the treatment without any coordination capabilities (none). These performance data suggest that the demes are ignoring messaging and instead are using their x and y coordinates to perform division of labor.

Logic Environment. Next, we examined how the demes performed division of labor in the logic environment. We performed the same three additional experimental treatments. In this case, the performance of the various treatments were not qualitatively different (data not shown). This led us to uncover a third possible mechanism for assigning roles: using random information to evolve a stochastic strategy. The organisms could acquire random information using either their cell data or the inputs to their logic operations. While this is a clever strategy, we were more interested in which mechanism they would use if randomness was not available. Thus, to force the organisms to evolve a non-stochastic strategy, we deprived them of both sources of random information and repeated all four treatments.

Figure 7 depicts the mean (A) and maximum (B) performance of the demes for these treatments that exclude random information averaged across 30 runs. In these plots, the location & messaging treatment differs from the original 9-logic treatment (depicted in Figure 5) in that random information has been removed. Interestingly, the location & messaging treatment that does not include random information significantly outperforms the original treatment that does. The mean and maximum deme performance are 5.44 (se±0.10) and 8.27 (se±0.14) tasks for the location & messaging treatment, as compared to 4.27 (se±0.10) and 6.17 (se±0.16) tasks for the original treatment. This disparity indicates that random information presented an easier to evolve mechanism for division of labor that produced inferior results overall. Additionally, based on these treatments, we can see that when deprived of random information, the organisms use messaging, rather than location information to perform division of labor. The messaging only treatment matches the performance of the location & messaging treatment, both of which exceeded the performance of the location only treatment. These results differ from those of the 25 role environment in which organisms made use of location information rather than messaging capabilities.

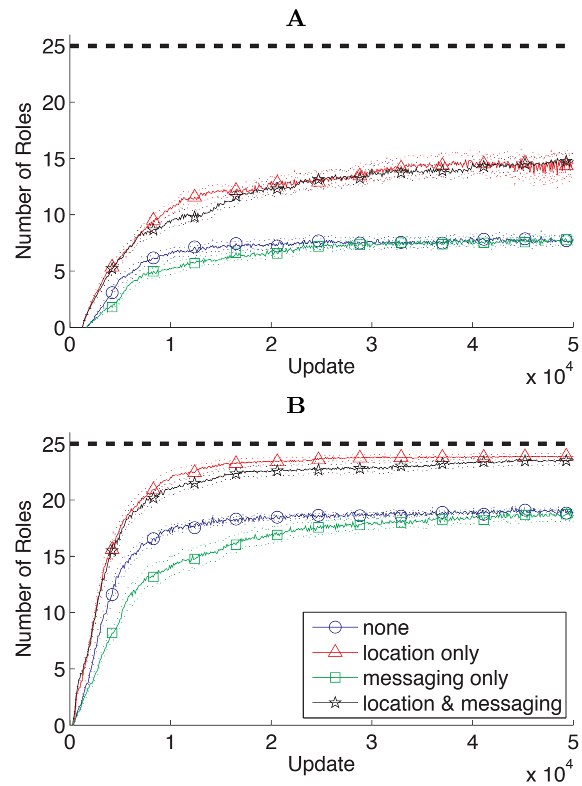

Figure 6: The mean (A) and maximum (B) number of roles performed in the 25 role environment when different combinations of instructions are removed. For this environment, deme performance depends upon the availability of location information.

Additional Experimental Treatments. We performed several additional experiments to assess the generality of our results. First, we performed versions of the 25 role treatments in which we deprived the organisms of random information to verify that this factor was not critical to the evolution of their strategy. The results were qualitatively similar; the organisms preferred to use location information, rather than their messaging capabilities. Second, we conducted additional treatments designed to assess if we were biasing our results by only providing three mechanisms for specialization. These new treatments enabled the organisms to perform temporal specialization by providing instructions that access the number of cycles the organism had executed (i.e., age of the organism) or the number of updates since

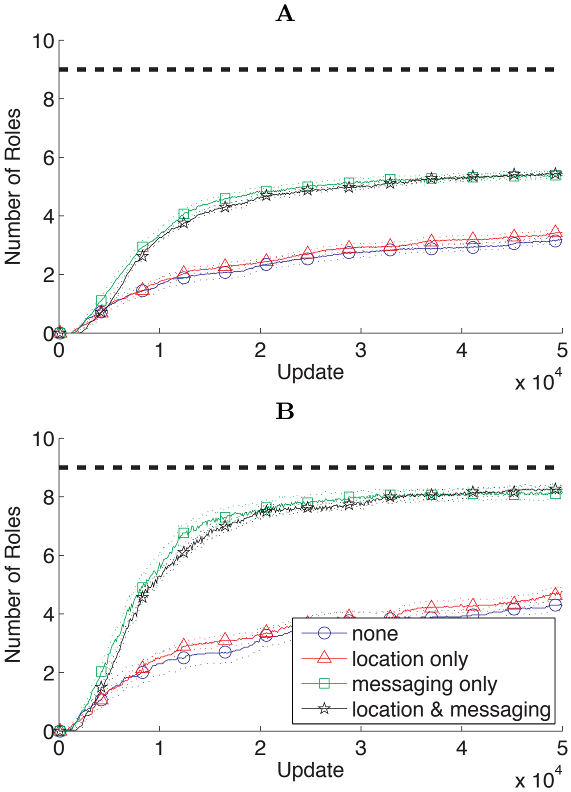

Figure 7: The mean and maximum number of roles performed in the logic environment when different combinations of instructions are removed. For this environment, deme performance depends upon the availability of messaging capabilities.

the last deme replication event (i.e., age of the deme). However, including these instructions did not improve the performance of the demes (data not shown). These results indicate that the demes prefer to use messaging capabilities and location information to perform division of labor.

One open question is why the mechanism used by the demes differed between environments. We identified two possible hypotheses: The first hypothesis is that the type of task influences which coordination mechanism is preferred. The second hypothesis is that the number of tasks influences which coordination mechanism is preferred. Because the two experiments differ both in terms of number and type of task, we needed to perform an additional experiment to identify which hypothesis might explain the variation.

To test these hypotheses, we created a variation of the 25 role environment that had only nine different roles. Organisms could fill these roles by setting their role-ids to the numbers 1-9. This environment used the same type of tasks as the 25 role environment, but had the same number of tasks as the logic environment. If the demes evolved to use messaging, this would falsify our hypothesis that the type of task affected the mechanism of division of labor. Conversely, if the demes evolved to use location information, this would falsify our hypothesis that the number of tasks is the key factor in selecting a mechanism for division of labor. Figure 8 depicts the mean (A) and maximum (B) performance of the demes for these treatments. The mean performance of the treatments indicates that the organisms are using location information, rather than their messaging capabilities. This result provides evidence that it is the type of task, rather than the number of tasks, that was the key factor in selecting which mechanism to use. One possible explanation for this behavior is that in order to perform the role-based tasks, the organisms need to generate numbers representing their role-ids. Location information provides numbers that can be used as inputs for an algorithm that generates distinct role-ids.

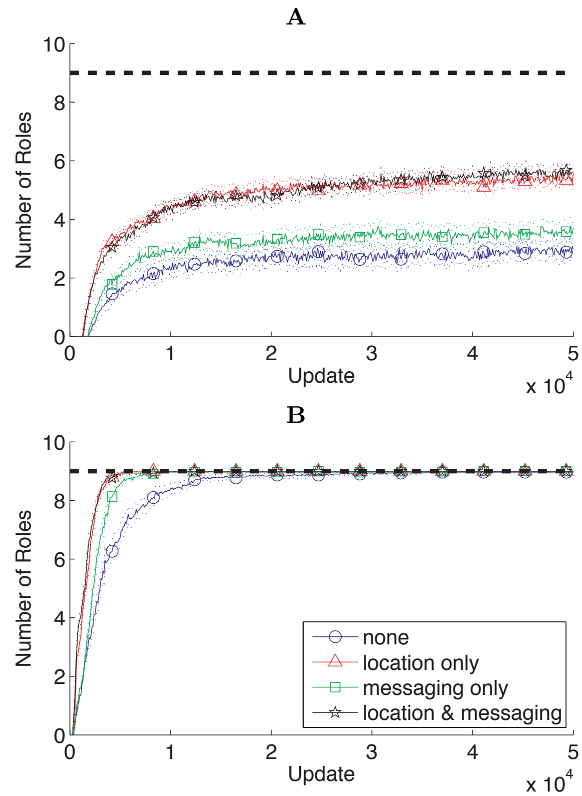

Figure 8: The mean (A) and maximum (B) number of roles performed in the 9 role environment when different combinations of instructions are removed. For this environment, deme performance depends upon the availability of location information indicating that the type of tasks is more important than the number of tasks.

4.3 Under what environmental conditions is a division of labor strategy preferred?

In the previous experimental treatments, we required organisms to specialize on performing a role. Within the 25 role environment, this behavior evolved as a byproduct of us polling which roles were being performed in each deme at a specific point in time and replicating those demes with the largest set of roles. Within the logic environment, the tasks were mutually exclusive and thus the organism could only ever perform one type of task (i.e., the type of the first task that they performed). In our final set of experiments, we release this constraint and enable organisms to continuously vary from being pure generalists that perform all of the different types of tasks to pure specialists that perform one type of task.

We refer to our new environment as the *resource consumption environment*. The objective for demes in this environment is to consume 350 units of resource as quickly as possible. Organisms within a deme can consume resources by performing any of five different types of logic tasks. As soon as the requisite amount of resources have been consumed, the deme is replicated, replacing another randomly selected deme. By consuming resources more rapidly and thus replicating more rapidly, a deme is better able to survive.

We configured the experimental environment such that a deme that performed all five types of logic tasks would consume resources more rapidly. Specifically, the environment is configured with 100 initial resources for each task with an inflow rate of 2 units per update. When an organism performs a task, it can consume up to 5% of the available resources associated with that type of task. Given these parameters, the most efficient way for a deme to replicate is to perform each of the 5 tasks 25 times. Optimally, this takes approximately 1 update. By contrast, if a deme only performs a single type of task, it takes approximately 130 updates to replicate. To start these experiments, we seeded all demes with 25 organisms that perform one logic task.

Our experiments test the hypothesis that as the cost of switching tasks increases, division of labor is more likely to evolve. Specifically, an organism is punished with a penalty (measured in terms of extra cycles) each time it switches the type of task it is performing. We experiment with costs of 0, 5, 10, and 50 cycles. Without task switching penalties each organism in the deme could be a generalist and perform each type of task once – resulting in each type of task being done 25 times total. With task switching penalties, demes would be more efficient if organisms specialized in performing one type of task 5 times. If five organisms specialized on each task, the deme would still perform each task 25 times.

Each treatment was replicated 30 times. To ensure that higher costs were not deleteriously affecting the overall performance of the demes, we examined *gestation cycle length*, the average number of updates needed for a deme to replicate. There was not an appreciable difference between the cost and no-cost treatments, which indicated that the increasing costs did not prevent the demes from being replicated. Next, we examined the affect of cost on task switching. The organisms within a replicating deme with cost 0, 5, 10, and 50 cycles switched tasks 13.28 (se$\pm$1.48), 10.00 (se$\pm$1.10), 8.50 (se$\pm$0.65), and 2.44 (se$\pm$0.42) times, respectively. As the penalty for task switching increases, the number of task switches per organism decreases. Notably, in the 50 cycle cost treatment, 11 of the 30 runs achieved near perfect specialization where the average number of switches per organism was appreciably less than 1, and one run achieved perfect specialization, where the number of switches was exactly 0.

Analysis. To better understand how demes evolved to perform division of labor, we analyzed one successful deme in more detail. Specifically, we chose to examine one of the demes from the 50-cyle cost treatment, where all of the demes had evolved to consume 350 resources without incurring any task switching penalties. This deme performed 4 different tasks and had an average deme gestation time of less than 2 updates.

```
...
nop-A
get-role-id          # BX <- latest role-id
set-flow
push
retrieve-msg         # (CX,AX) <- (label,data)
nop-C
rotate-right-one
rotate-left-one
nand                 # CX <- BX nand CX ~(role-id & label)
nop-C
if-label             # always true
if-less              # if BX < CX ~(role-id & nand value)
dec                  #   --BX
bcast1               # bcast BX,CX
IO                   # IO CX
nop-C
...
```

Figure 9: A snippet of the genome of a organism that when placed with copies of itself in a deme performs division of labor and problem decomposition.

The first step in our analysis was to identify which mechanism was being used to perform division of labor. To identify the mechanism used, we conducted “knockout” experiments for each possible coordination instruction, including messaging capabilities, location information, age of the organism, and age of the deme. In each knockout experiment, all instances of the target instruction were replaced with a placeholder instruction that performs no useful function. We then seeded a deme with the knockout version of the genotype and assessed its performance. The deme’s performance was not affected by the removal of location information, age of the organism, or age of the deme. However, if the deme’s messaging capability was disabled, it was no longer able to perform any tasks, and was thus unable to consume any resources or replicate. By performing additional knockouts that removed only the send-msg instruction, we further confirmed that the contents of the message were an integral part of the deme’s strategy.

The second step was to isolate how messaging was being used by the deme. Figure 9 lists the relevant fragment of the genome. Essentially, an organism uses three pieces of information to compute the results of a task: its role-id register (which serves as a place to store information), the label of a message it receives, and the contents of a message it receives. At a high level, an organism NANDs together information it receives from another organism with information it stores in its role-id register to perform a task.

One fascinating element of this strategy is that the organisms are using messaging, not just as a mechanism for regulating task assignment, but also as a means to cooperatively develop solutions for more complex tasks: Some organisms performed the simple tasks. They then sent messages to other nearby organisms containing the results of these tasks. The nearby organisms performed the more complex tasks using the simple task solutions sent to them through messaging. This building block strategy requires organisms all have the same non-random inputs. To verify this strategy element, we turned on random information. With random information demes of organisms were only able to perform the 2 easiest logic tasks indicating the organisms are sharing information to perform more complex tasks. This deme of organisms not only evolved to perform division of labor, but also evolved to perform cooperative problem decomposition.

5. CONCLUSIONS

In this paper, we explored the evolution of division of labor within clonal demes. Specifically, we demonstrated that demes of digital organisms were able to evolve to engage in division of labor in a variety of environments. The mechanism that organisms used to select a role to perform varied by the type of environment. In the 25 role and 9 role environments, the organisms used location information. In the logic environment, the organisms used messaging. Lastly, we examined the role of task switching costs on the evolution of division of labor. As task switching costs increase, organisms specialize more, resulting in division of labor. One deme that exhibited perfect division of labor used messaging as its mechanism and also exhibited an emergent form of problem decomposition. In future work, we seek to use this technique to better understand the behavior of social organisms and to harness this approach to apply evolutionary computation to complicated problems that may require problem decomposition. For example, in previous work we evolved organisms that represented software models [14], but were limited by the number and complexity of components that could be represented within one organism's genome. In future work, we could use division of labor to generate a group of organisms, each of which represent a model of a unique software component, that cooperate to solve the overall software engineering problem.

6. REFERENCES

[1] J. C. Bongard. The legion system: A novel approach to evolving hetrogeneity for collective problem solving. In *Proceedings of the European Conference on Genetic Programming*, pages 16–28, London, UK, 2000. Springer-Verlag.

[2] J. T. Bonner and K. B. Raper. A theory of the control of differentiation in the cellular slime molds. *The Quarterly Review of Biology*, 51(50th Anniversary Special Issue, 1926-1976):296–312, 1976.

[3] B. J. Crespi. The evolution of social behavior in microorganisms. *Trends in Ecology & Evolution*, 16(4):178–183, April 2001.

[4] D. C. Dennett. The new replicators. In M. Pagel, editor, *The Encyclopedia of Evolution*, volume 1, pages E83–E92. Oxford University Press, 2002.

[5] A. Dornhaus, F. Klügl, F. Puppe, and J. Tautz. Task selection in honeybees - experiments using multi-agent simulation. In *Proc. of the Third German Workshop on Artificial Life*, 1998.

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

[7] D. Floreano, S. Mitri, S. Magnenat, and L. Keller. Evolutionary Conditions for the Emergence of Communication in Robots. *Current Biology*, 17:514–519, 2007.

[8] S. Goings and C. Ofria. Ecological approaches to diversity maintenance in evolutionary algorithms. In *IEEE Symposium on Artificial Life*, Nashville, TN, 2009.

[9] J. M. Jandt and A. Dorhaus. Spatial organization and division of labour in the bumblebee bombus impatiens. *Animal Behavior*, 77:641–651, 2009.

[10] D. B. Knoester, P. K. McKinley, and C. Ofria. Cooperative network construction using digital germlines. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2008.

[11] T. H. Labella, M. Dorigo, and J.-L. Deneubourg. Division of labor in a group of robots inspired by ants' foraging behavior. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 2006.

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

[13] P. Lichodzijewski and M. I. Heywood. Managing team-based problem solving with symbiotic bid-based genetic programming. In *Genetic and evolutionary computation conference (GECCO)*, pages 363–370, Atlanta, Georgia, 2008.

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

[15] C. Ofria and C. Adami. Evolution of genetic organization in digital organisms. In *Proc. of DIMACS workshop Evolution as Computation*, pages 167–175, Princeton, NJ, 1999.

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

[17] N. Pike and W. A. Foster. *Ecology of Social Evolution*. Springer Berlin Heidelberg, 2008.

[18] M. A. Potter and K. A. DeJong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evol. Comput.*, 8(1):1–29, 2000.

[19] D. C. Queller and J. E. Strassmann. Eusociality. *Current Biology*, 13(22):R861–863, 2003.

[20] G. Robinson. Regulation of division of labor in insect societies. *Annual Review of Entomology*, 37:637–665, 1992.

[21] M. Waibel, D. Floreano, S. Magnenat, and L. Keller. Division of labour and colony efficiency in social insects: effects of interactions between genetic architecture, colony kin structure and rate of perturbations. *Proceedings of the Royal Society B*, 273:1815–23, 2006.

[22] M. Waibel, L. Keller, and D. Floreano. Genetic team composition and level of selection in the evolution of cooperation. *IEEE Transactions on Evolutionary Compuation*, 2009 (To appear).

[23] E. O. Wilson. Caste and division of labor in leaf-cutter ants (hymenoptera: Formicidae: Atta): I. the overall pattern in a. sexdens. *Behavioral Ecology and Sociobiology*, 7(2):143–156, 1980.

[24] C. H. Yong and R. Miikkulainen. Cooperative coevolution of multi-agent systems. Technical report, University of Texas at Austin, 2001.