

THE EVOLUTION OF DIVISION OF LABOR IN DIGITAL ORGANISMS

By

Heather J. Goldsby

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Computer Science
Ecology Evolutionary Biology, & Behavior

2011

ABSTRACT

THE EVOLUTION OF DIVISION OF LABOR IN DIGITAL ORGANISMS

By

Heather J. Goldsby

Division of labor is a hallmark strategy employed by a wide variety of groups ranging in complexity from bacteria to human economies. Within these groups, some individuals, such as worker ants, sacrifice their ability to reproduce and instead dedicate their lives to the maintenance of the colony and success of their kin. A worker ant may spend its entire life performing a single task, such as defending the colony or tending to the brood. The complexity of the strategies employed by these groups, combined with their rampant success, gives rise to questions regarding why division of labor exists. While extensive research has been done to better understand the patterns and mechanisms of division of labor, exploring this topic in an evolutionary context remains challenging to study due to the slow pace of evolution and imperfect historical data.

Understanding how and why division of labor arises is pertinent not just for understanding biological phenomena, but also as a means to enable evolutionary computation techniques to address complex problems using problem decomposition. The objective of problem-decomposition approaches is to have a group of individuals cooperatively solve a complex task by breaking it into pieces, having specialist individuals solve the pieces, and reassembling the solution. Essentially, problem-decomposition approaches use division of labor to enable groups to solve more challenging problems than any individual could alone. Unfortunately, human engineers have struggled with creating effective, automated problem-decomposition approaches.

In this dissertation, I use digital evolution (i.e., populations of self-replicating computer

programs that undergo open-ended evolution) to investigate questions related to the evolution of division of labor and to apply these insights to problem decomposition techniques. This dissertation has three primary components: First, we provide experimental evidence that evolutionary computation techniques can evolve groups of individuals that exhibit division of labor. Second, we explore two hypotheses for the evolution of division of labor. Specifically, we find support for the hypothesis that temporal polyethism (i.e., where a worker's age is related to the task it performs within the colony) may result from the evolutionary pressures of aging and risks associated with tasks. Additionally, we find support for a hypothesis initially proposed by Adam Smith, the premier economist, that the presence of task-switching costs results in an increase in the amount of division of labor exhibited by groups. Third, we describe how our analyses revealed that groups of organisms evolved as part of our task-switching work exhibit complex problem decomposition strategies that can potentially be applied to other evolutionary computation challenges. This work both informs biological studies of division of labor and also provides insights that can enable the development of new mechanisms for using evolutionary computation to solve increasingly complex engineering problems.

Copyright by
HEATHER J. GOLDSBY
2011

For Dave.

Thank you for your never-ending love, support, encouragement, and sense of adventure as we tackle challenges together in both the virtual and physical world. It's nice to be done climbing Half Dome. I can't wait to see what we decide to conquer next.

Acknowledgements

I am thankful for the myriad of people who have contributed directly or indirectly to my Ph.D. studies. First, I would like to thank my family: Doug and Jill Goldsby, Kelley and Matt Davis, Kurt and Julie Goldsby, Dave and Marcia Knoester, Brett and Alexandria Filush, and Jenny Knoester for their support throughout this process. I especially appreciate the effort all of them have made to understand my research.

The wisdom and advice of my committee members has made my research journey an enjoyable process. I would like to thank Dr. Charles Ofria, my Ph.D. advisor, for his excitement over science and research, which proved to be contagious, his dedication to finding innovative means to address challenging questions, and his insights that greatly contributed to this work. Additionally, I would like to thank Dr. Betty H. C. Cheng, my initial Ph.D. advisor, for teaching me the nuts and bolts of writing, researching, and presenting. I very much appreciate her mentorship and friendship over the years. Furthermore, I wish to thank my other committee members for their invaluable contributions and advice: Dr. Erik Goodman, Dr. Fred Dyer, and Dr. Philip McKinley.

Throughout my studies, I have had the distinct pleasure of being a member of the Digital Evolution Laboratory (Devolab) and Software Engineering and Network Systems (SENS) Laboratory. I am grateful for numerous contributions that members of these laboratories have made to my work. I would especially like to thank Sascha Konrad for his assistance in writing my first few papers, Sherri Goings and Jeff Clune for easing my transition into digital evolution research, and Dave Knoester for his assistance in my fledgling software development efforts.

TABLE OF CONTENTS

List of Tables	ix
List of Figures	x
1 Introduction	1
1.1 Biological Background and Motivation	2
1.2 Evolutionary Computation Background and Motivation	3
1.3 Approach	5
1.4 Thesis Statement	6
1.4.1 Direct Selection For Division of Labor	6
1.4.2 The Evolution of Division of Labor	8
1.4.3 Applications of Division of Labor: Problem Decomposition	9
1.5 Contributions	10
2 Methods	11
2.1 Avida	11
2.1.1 Instructions	13
2.1.2 Tasks and Resources	13
2.1.3 Groups	15
2.1.4 Genetically Heterogeneous Groups	16
2.1.5 Genetically Clonal Groups	17
2.1.6 Experimental Configuration	19
2.2 Measuring division of labor	19
3 Direct Selection for Division of Labor	21
3.1 Leaders and Followers: Frequency-Dependent Selection	21
3.1.1 Methods	24
3.1.2 Results	29
3.1.3 Discussion	33
3.2 Leaders and Followers: Multilevel Selection	34
3.2.1 Methods	34
3.2.2 Results	35
3.2.3 Discussion	37
3.3 Division of Labor: Heterogeneous Groups	39
3.3.1 Methods	39
3.3.2 Results	39
3.3.3 Analyses	46
3.3.4 Discussion	59

3.4	Division of Labor: Clonal Groups	59
3.4.1	Methods	60
3.4.2	Results	60
3.4.3	Discussion	68
4	Forces Driving the Evolution of Division of Labor	70
4.1	Temporal Polyethism	70
4.1.1	Methods	71
4.1.2	Results	73
4.1.3	Analyses	80
4.1.4	Discussion	88
4.2	Task-Switching Costs	89
4.2.1	Methods	89
4.2.2	Results	91
4.2.3	Analyses	100
4.2.4	Discussion	104
5	Applications of Division of Labor: Problem Decomposition	107
5.1	Engineered Problem Decomposition Approach	107
5.1.1	Methods	109
5.1.2	Results	115
5.1.3	Conclusions	126
5.2	Evolved Problem Decomposition Approach	127
5.2.1	Group case study 1.	127
5.2.2	Group case study 2.	129
5.3	Implications	131
6	Conclusions	133
6.1	Contributions	133
6.2	Future Work	135
6.2.1	Problem Decomposition in Embodied Agents	135
6.2.2	Evolution of Division of Labor	136
6.2.3	Major Transitions in Evolution	138
	APPENDICES	141
A	Overview of Additional Research Projects	142
A.1	Avida-MDE	143
A.2	Marple	145
A.3	Conclusions	148
	LIST OF REFERENCES	151
	BIBLIOGRAPHY	151

LIST OF TABLES

2.1	Coordination instructions for this study. These instructions are in addition to the general Avida instruction set.	13
3.1	The payoff matrix for the leader-follower game, where a is the multiplicative benefit of leadership, b is the coordination benefit, and c is the cost of leadership. In general, leaders have higher payoffs if the dyad coordinates roles, but a lower payoff if they do not.	25
3.2	Alternative hypotheses and predictions tested in this study.	48
4.1	The four risk treatments for a two-task environment. The rows describe the lethality risks associated with tasks not and nand . (E.g., A 25% risk means that while performing the task, the organism has a 25% chance of being killed prior to receiving credit for the task.) The columns describe a specific treatment.	73
4.2	The eight risk treatments for a three-task environment. The rows describe the lethality risks associated with tasks not , nand , and ornot . The columns describe a specific treatment.	80
5.1	Additional Avida instructions that allow organisms to produce strings, donate strings, and selectively cooperate with neighbors on the basis of reputation.	113

LIST OF FIGURES

2.1	Left: An Avida population containing multiple genotypes in a spatial environment, where different colors indicate organisms with different genotypes. Right: The structure of an individual organism, including its genome, input, internal registers, and output. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.	12
2.2	Heterogeneous Group Replication Process: (1) An Avida population containing four heterogeneous groups. Group A completes a group task or wins a group competition. (2) At this point, group B is selected to be replaced and all its organisms are removed. (3) A perfect copy of group A is created and is loaded into group B.	17
2.3	Clonal Group Replication Process: (1) An Avida population containing four clonal groups. Group A completes a group task or wins a group competition. (2) At this point, group B is selected to be replaced and all its organisms are removed. (3) The germline of group A is used to create the germline of the new colony. (4) One individual from the new germline is placed into the target group. (5) The source group A is reset to its original state with one individual.	18
2.4	A depiction of the amount of Shannon mutual information present among a group of tasks and a group of individuals. (a) All individuals specialize on performing the same task resulting in a low amount of division of labor. (b) A generalist individual performs all of the tasks resulting in a low amount of division of labor. (c) Each individual specializes on performing a different task resulting in a high amount of division of labor.	20
3.1	A screenshot of the leader-follower game developed using NetLogo and available online. The game enables the user to configure the costs of leadership, benefits of leadership, and the benefits of division of labor.	28

3.2	The game-theoretic simulation results for various combinations of costs and benefits. The red line represents the percentage of the population that selected the leader strategy and the blue line represents the percentage of the population that selected the follower strategy. In the first treatment (a), leadership benefit is 3, coordination benefit is 1, and leadership cost is 0. These values result in many leaders. In the second treatment (b), leadership benefit is 5, coordination benefit is 1, and leadership cost is 2. These values result in an equal number of leaders and followers. These results demonstrate that frequency-dependent selection can be used to produce various population compositions of leaders and followers even if the payoffs per interaction are not equal.	30
3.3	The simulation results for various combinations of probabilistic costs given the same division of labor benefit, leadership benefit, and mean cost. The red line represents the percentage of the population that selected the leader strategy and the blue line represents the percentage of the population that selected the follower strategy.	32
3.4	The mean number of leaders (a) and followers (b) when each group's target is one leader and 24 followers. The dashed line represents the target value in each case. Each line represents a different treatment that has a different multiplicative bonus value of leadership varying from 0.5 to 64. Overall, the group-level component of the multilevel selection pressure is sufficient to produce groups with few leaders and many followers, despite the appreciable pressure for individuals to be leaders.	36
3.5	The mean number of leaders (a) and followers (b) when each group's target is 13 leaders and 12 followers. The dashed line represents the target value in each case. Each line represents a different treatment that has a different multiplicative bonus value of leadership varying from 0.5 to 64. Overall, the group-level component of the multilevel selection pressure is sufficient to produce groups with few leaders and many followers, despite the appreciable pressure for individuals to be leaders.	37
3.6	The results of using multilevel selection to evolve groups that comprise varying numbers of roles, where each line represents a different treatment with a different desired number of roles ranging from 2 to 20. The grand mean (a) and mean of the maximum number of unique roles (b) over all 30 trials. In general, the groups evolved to perform the desired number of roles.	42

3.7	The results of using just individual selection to evolve groups that comprise varying numbers of roles, where each line represents a different treatment with a different desired number of roles ranging from 2 to 20. In these control treatments, the group-level component of multilevel selection was removed. Within this figure, (a) depicts the grand mean and (b) depicts the mean of the maximum number of unique roles over all 30 trials. Without the group-level component of the multilevel selection pressure, the groups were unable to evolve to perform multiple roles on average and in the best case evolve to perform 7 roles.	44
3.8	Control and experimental results of using multilevel selection pressures to evolve groups of organisms that perform five different logic tasks with varying rewards. The dashed line represents the best possible performance. Here, (a) depicts the grand mean number of roles performed and (b) depicts the mean of max roles performed across 30 trials. The two control treatments (No group; no ind and No group; ind) each evolve to perform fewer than 3 roles; whereas, the experimental treatment with both components of the multilevel selection pressure evolves to perform 4 roles on average and 5 at best. These results are evidence that multilevel selection pressure is sufficient to produce groups that perform five logic tasks with varying rewards.	47
3.9	The average mean (a) and average maximum (b) performance of the groups through the normal 50,000 update period and then a 10,000 update ecological phase, where mutations are turned off. The horizontal dashed line represents the maximum number of tasks possible and the vertical dashed line represents the start of the ecological period. The dotted lines represent standard error. The mean group performance increases and the maximum group performance does not decrease from 50,000-60,000 updates, indicating that mutations are not a key part of the group strategy.	49
3.10	The percent of genetic diversity in the dominant Logic-5 groups and control runs. Notably, the Logic-5 groups have more genetic diversity even though they also must maintain this diversity while performing tasks with different associated benefits.	54
3.11	The 15 genotypes present in group 29 after an ecological period of 100 updates. Each column is one genotype. Each column element is an instruction. The coordination instructions are colored. There are two different ecotypes present within this group. They are distinguishable by the different patterns of banding.	57

3.12	A visual depiction of the genotype and phenotypes of the organisms that comprise group 29. The shading represents the task performed (phenotype). The first number represents the genotype and the second number represents also denotes the task performed (phenotype). The black squares are the organisms in ecotype family 2. The remainder of the squares are in ecotype family 1. .	58
3.13	The mean and maximum number of roles performed by groups within the 25-role environment across 30 trials. Notably, some groups approach a perfect score (denoted by a dashed line) of 25.	62
3.14	The mean and maximum number of roles performed by groups in the logic environment, where there are nine tasks of increasing complexity. The best possible performance is denoted with the dotted line. Groups on average perform four tasks and at best perform 6 tasks.	63
3.15	The mean (a) and maximum (b) number of roles performed in the 25-role environment when different combinations of coordination instructions are removed. For this environment, group performance depends upon the availability of location information.	65
3.16	The grand mean (a) and mean of maximum (b) number of roles performed in the logic-9 environment when different combinations of instructions are removed. For this environment, group performance depends upon the availability of messaging capabilities.	67
3.17	The mean (a) and mean of maximum (b) number of roles performed in the 9-role environment when different combinations of instructions are removed. For this environment, group performance depends upon the availability of location information indicating that the type of tasks is more important than the number of tasks.	69
4.1	The ancestor organisms for two-task temporal polyethism experiments. The not-nand ancestor performs task not , performs task nand , and then replicates. The nand-not ancestor performs task nand , performs task not , and then replicates. Because the genomes are circular, after each organism replicates it resumes execution at the top of its genome.	74

4.2	The results of the temporal polyethism treatments in which task not is risky. The x-axis of both plots is evolutionary time and the y-axis is the mean age in cycles. The blue line with circles represents the mean age at which task not is performed. The red line with triangles represents the mean age at which task nand is performed. In these results, the more risky task is performed later in the lifetime of the organism, which is evidence for our hypothesis that task riskiness may be an evolutionary pressure that gives rise to a temporal polyethism pattern.	75
4.3	The results of the temporal polyethism treatments in which task nand is risky. The x-axis of both plots is evolutionary time and the y-axis is the mean age in cycles. The blue line with circles represents the mean age at which task not is performed. The red line with triangles represents the mean age at which task nand is performed. In these results, the more risky task is performed later in the lifetime of the organism, which is evidence for our hypothesis that task riskiness may be an evolutionary pressure that gives rise to a temporal polyethism pattern.	76
4.4	Diagrams of the results of the control temporal polyethism treatments in which neither task is risky. The x-axis of both plots is evolutionary time and the y-axis is the mean age in cycles. The blue line with circles represents the mean age at which task not is performed. The red line with triangles represents the mean age at which task nand is performed. In these results, the controls indicate that there is nothing intrinsic about the tasks that is driving the temporal polyethism results.	78
4.5	Diagrams of the results of the control temporal polyethism treatments in which both tasks are risky. The x-axis of both plots is evolutionary time and the y-axis is the mean age in cycles. The blue line with circles represents the mean age at which task not is performed. The red line with triangles represents the mean age at which task nand is performed. In these results, the controls indicate that there is nothing intrinsic about the tasks that is driving the temporal polyethism results.	79
4.6	The six ancestor organisms for the three-task temporal polyethism experiments. These ancestor organisms vary the order in which the three tasks are performed. Because the genomes are circular, after each organism replicates, it resumes execution at the top of its genome.	81

4.7	The results of the temporal polyethism experimental treatments for the not-nand-ornot ancestor in which task not is safe. The blue line with circles represents task not , the red line with triangles represents task nand , and the green line with squares represents task ornot . The x-axis is evolutionary time and the y-axis is the mean age at which the task is performed by the organisms. At the end of the treatments, the least risky task (0%) is performed first, the somewhat risky task (7%) is performed next, and the most risky task (15%) is performed last.	82
4.8	The results of the temporal polyethism experimental treatments for the not-nand-ornot ancestor in which not is moderately risky. The blue line with circles represents task not , the red line with triangles represents task nand , and the green line with squares represents task ornot . The x-axis is evolutionary time and the y-axis is the mean age at which the task is performed by the organisms. At the end of the treatments, the least risky task (0%) is performed first, the somewhat risky task (7%) is performed next, and the most risky task (15%) is performed last.	83
4.9	The results of the temporal polyethism experimental treatments for the not-nand-ornot ancestor in which not is extremely risky . The blue line with circles represents task not , the red line with triangles represents task nand , and the green line with squares represents task ornot . The x-axis is evolutionary time and the y-axis is the mean age at which the task is performed by the organisms. At the end of all of the treatments, the least risky task (0%) is performed first, the somewhat risky task (7%) is performed next, and the most risky task (15%) is performed last.	84
4.10	The results of the two temporal polyethism control treatments for the not-nand-ornot ancestor. The blue line with circles represents task not , the red line with triangles represents task nand , and the green line with squares represents task ornot . The x-axis is evolutionary time and the y-axis is the mean age at which the task is performed by the organisms. The ages at which the tasks are performed are not significantly different.	85
4.11	These results depict the mean age at which task not (blue line with circles), task nand (red line with triangles) and replication (black line with stars) are performed for the treatment where not is risky and the runs were started with the not-nand ancestor. These results suggest that the organisms are performing task nand one or more times, replicating, and then performing task not	86

4.12	Diagrams of the phenotype (left) and genotype (right) of an organism whose group exhibited temporal polyethism with two tasks. The numbered arrows surrounding the genotype indicate the order in which instructions were executed to produce the phenotype. In this case, the genotype is very similar to the not-nand ancestor. The risk-based order in which the tasks were performed depended upon control-flow instructions in the genome.	87
4.13	Diagrams of the phenotype (left) and genotype (right) of an organism whose group exhibited temporal polyethism with three tasks. The numbered arrows surrounding the genotype indicate the order in which instructions were executed to produce the phenotype. In this case, the genotype of the not-nand-ornot ancestor has been rearranged to facilitate temporal polyethism. Specifically, not and nand have exchanged places and an additional not has been introduced. The risk-based order in which the tasks were performed depended upon both the genomic architecture and the control-flow instructions in the genome.	88
4.14	Logic-9 environment results: (a) The mean Shannon mutual information between organisms and tasks performed averaged across 50 runs for groups with varying amounts of task-switching costs. Dotted lines are used to indicate standard error. Notably, treatments with higher task-switching costs evolve strategies that exhibit higher levels of division of labor. (b) The mean number of different types of tasks performed by the groups under various treatments. The groups typically evolve to perform 5-6 tasks.	93
4.15	25-Role Environment Results: (a) The mean Shannon mutual information between organisms and tasks performed averaged across 50 runs for groups with varying amounts of task-switching costs within the 25-role environment. Dotted lines are used to indicate standard error. Notably, treatments with higher task switching costs evolve strategies that exhibit higher levels of division of labor. (b) The mean number of different types of tasks performed by the groups under various treatments. The groups all evolve to perform approximately 20 types of tasks.	96
4.16	Logic-9 and Role-6 Environment Results: (a) The mean Shannon mutual information between organisms and tasks performed averaged across 50 runs for groups with varying amounts of task-switching costs. Dotted lines are used to indicate standard error. Notably, treatments with higher task switching costs evolve strategies that exhibit higher levels of division of labor. (b) The mean number of different types of tasks performed by the groups under various treatments. The groups all evolve to perform approximately 10 types of tasks.	97

4.17	Logic-9 250 Resource Results: (a) The mean Shannon mutual information between organisms and tasks performed averaged across 50 runs for groups with varying amounts of task-switching costs. Dotted lines are used to indicate standard error. Notably, treatments with higher task-switching costs evolve strategies that exhibit higher levels of division of labor. (b) The mean number of different types of tasks performed by the groups under various treatments. The groups all evolve to perform 3-4 types tasks.	99
4.18	Logic-9 1000 Resource Results:(a) The mean Shannon mutual information averaged across 50 runs for groups with varying amounts of task-switching costs. Dotted lines are used to indicate standard error. (b) The mean number of different types of tasks performed by the groups under various treatments. The groups all evolve to perform 7-8 types of tasks.	101
4.19	Boxes depict how the amount of Shannon mutual information varies as different knockout treatments are performed for a group that exhibits division of labor in the Logic-9 environment. Notably, knocking out location does not significantly affect the behavior of the majority of the groups. However, knocking out messaging has a tremendous effect and lowers the median amount of division of labor.	104
4.20	Boxes depict how the amount of Shannon mutual information varies as different knockout treatments are performed for a group that exhibits division of labor in the 25-role environment. Notably, knocking out messaging does not significantly affect the behavior of the majority of the groups. However, knocking out location has a tremendous effect and lowers the median amount of division of labor.	105
4.21	The role-ids selected by the organisms in a successful group that performs 24 (out of 25 possible) different roles and exhibits division of labor in the 25-role environment. All but one id (0) are rewarded.	106
4.22	An excerpt of the genome used by a successful group evolved in the 25-role environment to perform 24 different roles. Coordination is achieved via a location-sensitive algorithm. Specifically, organisms set their role-ids to $x + 5 * y$	106
5.1	The infrastructure that allows Avida organisms to produce strings, donate strings, and cooperate using indirect reciprocity.	112

5.2	Number of organisms that produce complete sets (a) and the number of complete sets produced (b) with four different treatments that vary the tagging and reputation information available for the 16-bit variant. Populations where organisms have both sources of information significantly outperform populations that lack either.	118
5.3	The percentage of the population of organisms producing complete sets across the 4-bit and 8-bit string production problem variants. For simpler problems, generalists outperform specialists; whereas, for more complex problems, specialists outperform generalists. Mixed populations tend to perform at least as well as the isolated populations of generalists or specialists.	119
5.4	The percentage of the population of organisms producing complete sets across the 16-bit and 24-bit production problem variants. For simpler problems, generalists outperform specialists; whereas, for more complex problems, specialists outperform generalists. Mixed populations tend to perform at least as well as the isolated populations of generalists or specialists.	120
5.5	The number of complete sets produced by generalists, specialists, and mixed populations of both generalists and specialists for the 4-bit and 8-bit string production problem variants. For simpler problems, generalists outperform specialists; whereas, for more complex problems, specialists outperform generalists. Mixed populations tend to perform at least as well as the isolated populations of generalists or specialists.	121
5.6	The number of complete sets produced by generalists, specialists, and mixed populations of both generalists and specialists for the 16-bit and 24-bit string production problem variants. For simpler problems, generalists outperform specialists; whereas, for more complex problems, specialists outperform generalists. Mixed populations tend to perform at least as well as the isolated populations of generalists or specialists.	122
5.7	The population composition of mixed populations comprising both generalists and specialists for the 4-bit and 8-bit variants of the bit string production problem. In general, the more simple variants have a predominantly generalist population and the more complex variants have a predominantly specialist population.	124
5.8	The population composition of mixed populations comprising both generalists and specialists for the 16-bit and 24-bit variants of the bit string production problem. In general, the more simple variants have a predominantly generalist population and the more complex variants have a predominantly specialist population.	125

5.9	The percent of organisms that are generalists, specialists, and altruists. The percent of altruists in the population closely tracks the number of specialists in the population, indicating that generalists are not making donations. . . .	126
5.10	A snippet of the genome of a organism that when placed with clones of itself in a group performs division of labor and problem decomposition.	128
6.1	Illustration of fraternal and egalitarian transitions. Circles represent individuals, boxes represent groups, arrows represent steps. For a fraternal transition (a), an organism (1), produces two daughter organisms (2), these daughters choose to remain in a group (3), and eventually differentiate their behavior (4). For an egalitarian transition (b), two organisms with different phenotypes (1) are proximally near one another (2) and choose to form a group (3). Finally, further evolution differentiates the behavior of the organisms within the group (4).	139

Chapter 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 groups, including certain species of bacteria [16], slime molds [8, 16, 98], algae [81], multicellular organisms [111], eusocial insect, arthropod, and mammal colonies [16, 21, 53, 55, 98, 98, 131], and humans economies [32, 108]. The genetic composition of members of these groups range from clonal (e.g., slime molds, bacteria, and some eusocial insects) to genetically heterogeneous (e.g., the majority of eusocial insects, humans, and other mammals). Eusocial colonies exhibit some of the most remarkable examples of division of labor. 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 [55]. Moreover, many eusocial organisms, such as leaf-cutter ants [131], bumblebees [55], and aphids [92], 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 [55]. In this dissertation, we focus on understanding the evolution of

division of labor and applying these insights to the development of evolutionary computation techniques.

1.1 Biological Background and Motivation

Task-related division of labor has been proposed as one of central reasons for the success of eusocial insect colonies [53] and as such has been extensively studied by scientists (e.g., [8, 16, 55, 98, 131]). Among other results, scientists have identified several patterns of division of labor, including *temporal polyethism*, where the individual’s age predicts which task it specializes on [53, 102, 121]; *foraging for work*, where the organism’s location within the colony predicts the task specialization of the worker [31, 53, 55, 120, 107]; and *morphological specialization*, where the phenotype of the worker is correlated with the task it specializes on [131]. 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 [6, 101, 109]. However, these studies focus primarily on the mechanisms used to implement division of labor, rather than why division of labor strategies evolved [20].

Although exploring division of labor in an evolutionary context is challenging, several scientists have performed studies that investigate potential explanations for the evolution and maintenance of division of labor [15, 20, 53, 103]. Using natural systems, these scientists explore whether the potential adaptive benefits for division of labor are present within modern eusocial colonies. For example, Schmid-Hempel reviewed evidence that the structure of castes (either temporal or physical) within a colony is adaptive [103]. He concludes that there is insufficient evidence to demonstrate that caste structure is adaptive at an ecological scale, but that it may be adaptive at an evolutionary scale. Dornhaus tested the “jack-of-all trades; master of none” hypothesis that increased specialization results in more efficient workers for *Temnothorax albipennis*, which is an ant species that does not have morpholog-

ical castes [20]. She found that the degree of specialization does not predict efficiency. In fact, more specialized workers, in some cases, performed worse than their more generalist relatives. Couvillon and Dornhaus explored a second possible explanation for the adaptive significance of division of labor when they explored whether morphological differences in bumblebees (*Bombus impatiens*) could enable colony-level robustness to starvation [15]. In this case, they found that smaller workers, which are generally less efficient at all tasks compared to their larger sisters, are better able to survive a nectar shortage and thus help to ensure the survival of the colony.

However, studies that examine the current adaptive benefits of division of labor may not necessarily reveal the historical reasons for which division of labor was originally selected. Questions regarding the evolutionary conditions that initially give rise to division of labor are challenging, if not impossible, to study using natural systems due to imperfections in the historical data and long generation times. As the late evolutionary biologist John Maynard Smith [110] made the case: “So far, we have been able to study only one evolving system and we cannot wait for interstellar flight to provide us with a second. If we want to discover generalizations about evolving systems, we will have to look at artificial ones.”

1.2 Evolutionary Computation Background and Motivation

Evolutionary insights into how and why division of labor arises are pertinent not just for understanding biological phenomena, but also for enabling evolutionary algorithms (EAs) to address complex problems, such as developing ultra large-scale software systems [82] or teams of robots [7, 29, 62, 126, 134]. Traditionally, EAs contain a single population in which each individual encodes a complete solution to a problem. EAs work well for problems with a search space that can be reasonably well-explored, given the computational

resources available to the population. For example, in previous work, we used a more traditional EA to evolve software models, where each model was composed of a set of interacting state diagrams [37, 38, 40, 79]. However, as the complexity of the problems addressed using EAs increases, the size of the search space and the amount of time required to search for a satisfactory solution grows rapidly. For example, as we begin to expand the scope of the type of software we would like to evolve models for to include ultra large-scale systems [82], such as health-care systems and traffic-control system, which comprise multiple heterogeneous software components, the complexity may become too great for a single individual to encode. *Problem decomposition* has been proposed to address this limitation [94]. Problem-decomposition EAs decompose a problem into simpler subproblems, individuals produce *subcomponents* (where each subcomponent is a solution to a subproblem), and then the subcomponents are combined to obtain a solution to the overall problem [94]. Essentially, problem-decomposition EAs use division of labor, where individuals specialize on one aspect of the problem and cooperate to develop the overall solution, to solve more challenging problems than any individual could alone. One of the overarching challenges of problem-decomposition EAs is to have the population automatically and dynamically both decompose the problem (if necessary) and assemble a complete solution.

Numerous techniques have been proposed to automate the evolution of problem decomposition [14, 35, 75, 94, 117, 134]. In general, these techniques evolve specialists that produce subcomponents. The overall solution is considered to be a group of specialists comprising either one representative member of each species (e.g., [35, 94, 134]) or an EA-selected team (e.g., [14, 75, 117]).

Several approaches assemble the subcomponents as part of the fitness function [94, 134]. Specifically, the cooperative coevolution architecture [94] evolves two or more species in completely isolated populations. Cooperation between these species occurs at the time of fitness evaluation, when individuals from one species are evaluated with representatives from each

of the other species. The nature of these collaborations is determined by the user, as is the choice of representatives from each species, but the number of species is determined by the algorithm automatically. Additionally, Yong and Miikkulainen [134] apply the cooperative coevolution architecture to evolve coordinated predator behavior in a prey-capture task. Three Enforced SubPopulations [47] are created, each of which evolves an artificial neural network that represents the behavior for one predator [134]. During the fitness evaluation, a representative neural network from each subpopulation is placed in a simulated environment and the reward for captured prey is split equally among all representatives present. Unfortunately, these approaches are only applicable in problems where the general form of between-species collaborations is known *a priori*.

Other approaches enable the EA itself to dynamically assemble the subcomponents. These approaches are typically based on a market economy. For example, some learning classifier systems form teams of individuals that bid for their subcomponents to be used as part of the solution to the classification problem [75, 117]. Additionally, Cornforth and Kirley [14] propose a non-rule based approach to problem decomposition using a market-based model, where agents group together in a hierarchical fashion to form complex problem solutions. One limitation of this approach is that individuals do not dynamically evolve subcomponents of the problem, but rather innately represent preformed subcomponents.

1.3 Approach

Our research uses digital evolution to address both understanding the evolution of division of labor and also applying these insights to problem decomposition EAs. Specifically, we use Avida [87], a digital-evolution platform previously used within biology to study topics including the origin of complex features [72], evolutionary design of modularity [84], and the evolution of altruism [13, 34]. Avida has also been used within computer science to study

topics including evolving distributed behaviors [63, 64, 65, 66], energy conservation [5], and artifacts for self-adaptive systems [37, 38, 40]. 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 self-replicate. Avida satisfies Dennett’s three requirements for an evolutionary system (replication, variation [mutation], and differential fitness [competition]) [18], while enabling rapid open-ended evolution, precise experimental control, and data collection that does not influence the course of evolution.

1.4 Thesis Statement

Digital evolution studies provide insight into the evolution of division of labor and provide the basis for the development of problem-decomposition evolutionary algorithms.

1.4.1 Direct Selection For Division of Labor

Our early work addressed the foundational question of whether we could use evolutionary computation techniques to evolve groups of individuals that exhibit division of labor. Specifically, we addressed the following topics:

In collaboration with psychologists, we used game theory to explore potential mechanisms that could give rise to leader-follower systems that represent a two-role form of division of labor in which a leader receives a higher payoff than a follower. From an evolutionary perspective, this behavior is puzzling if the individuals are not related, since it may put a follower’s genes at an evolutionary disadvantage for survival. We discovered that this behavior could arise using frequency-dependent selection [46].

Within our game theoretic approach, individuals could only select to be leaders or followers. To better understand how groups evolved to exhibit division of labor, we implemented a leader-follower system in Avida. Within Avida, organisms had to evolve to perform tasks to denote their role and to coordinate with other members of their group. Leaders received a higher payoff than followers; however, groups were rewarded on the basis of how few leaders they elected with the twist that they were required to have at least one leader. In these studies, groups evolved division of labor strategies despite highly disparate reward structures in which leaders were rewarded up to 64 times as much as followers [43, 45].

We expanded on the results of our leader-follower study to see whether multilevel selection could give rise to genetically heterogeneous [43, 45] groups that perform more complex versions of division of labor that include both more roles and more challenging tasks. In these studies, an individual was required to be a specialist and was under pressure to perform the most highly-rewarded task(s). The group, however, required a diversity of specialists. Our initial questions focused on understanding how groups of organisms could coordinate roles, especially roles with disparate benefits. Within this system, we observed groups of individuals evolve heritable strategies for performing division of labor. We then examined an evolved strategy and demonstrated that a combination of genetic heterogeneity, ecological factors, and coordination mechanisms (such as physical condition and communication) were used to coordinate roles.

Next, to more thoroughly isolate questions of how groups perform division of labor, we turned our attention to genetically-clonal groups. We focused on understanding the mechanisms that groups of organisms used to coordinate roles (i.e., location awareness or communication) and how these mechanisms may be environmentally dependent [44]. We discovered that the mechanism varied according to the type of roles being performed.

These studies provided us with confidence that groups of organisms within Avida can indeed evolve to perform division of labor with complex tasks and unequal rewards.

1.4.2 The Evolution of Division of Labor

Given evolutionary computation can be used to produce groups that exhibit division of labor, our later work focused on understanding the conditions under which division of labor evolves as a strategy. For these studies, instead of directly rewarding for division of labor, we instead reward for group efficiency and observe when and how division of labor evolves. Specifically, we addressed the following topics:

Temporal polyethism, where a worker’s occupation changes with age, is one of the most common forms of division of labor [31, 53, 102, 107, 119, 120, 121]. An ongoing discussion within communities of division of labor researchers is why this pattern exists. One theory is that this behavior can be explained by the risks of aging and the relative risks of the tasks [53]. Specifically, more risky tasks are performed when the organism is older and closer to death. In this study, we used Avida to explore whether the risks associated with aging and different tasks are sufficient for evolving colonies that exhibit temporal polyethism. We test this hypothesis in both two-task and three-task scenarios and discover evidence that suggests aging and task riskiness are able to result in the evolution of a temporal polyethism pattern of division of labor.

Three advantages for human economic division of labor suggested by Adam Smith [108] and later proposed by Dornhaus [20] as viable explanations for division of labor within eusocial colonies are: (1) the efficiency with which an individual performs a task improves with repeated performances, (2) specialization can decrease task-switching costs, and (3) the creation of machines can assist organisms performing specialized tasks more efficiently. In these studies, we examine how task-switching costs (the second proposed benefit) affect whether or not division of labor evolves as a strategy [41, 44]. We discovered that the application of task-switching costs is sufficient to increase the amount of division of labor present within colonies of organisms. We then performed additional experiments to assess the generality of these results and found that they were robust to environmental perturbations

in terms of the criteria for successful colonies and the type of task within the environment. We also encountered support for the idea that efficiency gains through parallel processing may favor the evolution of division of labor strategies. While this was not the hypothesis we were testing, efficiency through parallel processing has previously been proposed as a benefit of division of labor [53, 88].

Within these studies, division of labor evolved as an efficient means to solve group problems. These studies address biological questions that are infeasible to experimentally study using natural systems.

1.4.3 Applications of Division of Labor: Problem Decomposition

The last aspect of our research focuses on whether these division of labor insights could be applied to problem decomposition approaches for EAs. Specifically, we addressed the following topics:

Prior to our division of labor studies, we engineered a problem decomposition EA that evolves specialists that produce subcomponents and then dynamically assemble the subcomponents to form a complete solution [42]. Unfortunately, our early technique relied upon developers to specify how the problem was to be decomposed and reassembled.

Analyses of the groups that evolved to perform division of labor as the result of task-switching costs revealed that they accomplished this feat using an advanced form of problem decomposition akin to strategies observed in nature, but not directly rewarded by our experiments [41, 44]. Specifically, (1) group members broke complex logic tasks into constituent simpler tasks, (2) individuals computed the results for those simpler tasks, (3) individuals then shared the task results, and (4) other neighboring individuals assembled the solutions to produce the results for more complex tasks. These studies form the basis for future work that will apply evolutionary pressures that result in division of labor to other problems that could benefit from problem decomposition including those in software engineering and the

design of robot-control algorithms.

1.5 Contributions

In summary, there are four main contributions of our research: (1) evidence that evolutionary computation techniques can evolve groups of individuals that exhibit division of labor, (2) support for the hypothesis that temporal polyethism may result from the evolutionary pressures of aging and risks associated with tasks, (3) support for the hypothesis that task-switching costs result in an increase in the amount of division of labor exhibited by groups, and (4) an example of how the evolutionary pressures that favor the evolution of division of labor may result in groups exhibiting problem decomposition.

In the following, we provide background information on the methods used for this research (Chapter 2). Next, we describe the results from the foundational studies (Chapter 3), elaborate on our studies of the evolution of division of labor (Chapter 4), and present our work on problem decomposition (Chapter 5). We conclude with the impact and future potential for this line of research (Chapter 6).

Chapter 2

Methods

In this chapter, we introduce Avida focusing on the portion of the infrastructure used for our experiments. Because this research addresses division of labor, we are especially interested in the interplay among individual organisms and their group. Additionally, we describe how we measure division of labor within groups.

2.1 Avida

For the majority of the work presented in this dissertation, we used the Avida digital evolution platform [87]. Figure 2.1 depicts an Avida population and the structure of an individual digital organism. An Avida population is composed of a number of *cells*, where each cell is a compartment in which an organism can live. A cell can contain at most one organism, and the total number of cells (and thus the size of an Avida population) is fixed. Cells are connected to each other via a configurable topology, generally a grid or torus, that defines the neighborhood of each cell. An organism’s *neighbors* are the organisms that live in the cells adjoining its own cell. Additionally, each organism has a *facing*, which is one connection to a neighboring cell. Organisms are able to change their facing by rotating.

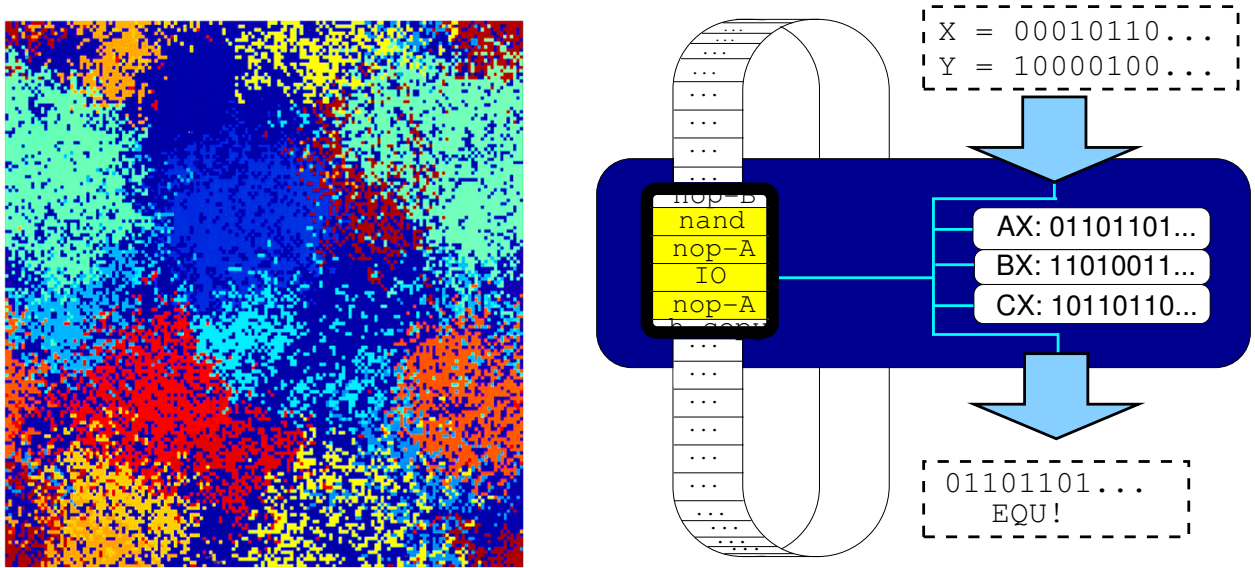


Figure 2.1: Left: An Avida population containing multiple genotypes in a spatial environment, where different colors indicate organisms with different genotypes. Right: The structure of an individual organism, including its genome, input, internal registers, and output. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

Each digital organism is defined by a circular list of instructions (its “genome”), a virtual CPU, and its position in a common virtual environment. The virtual CPU architecture comprises three general-purpose registers $\{AX, BX, CX\}$ and two stacks $\{GS, LS\}$. In Figure 2.1, each color within the population represents a different genotype. An organism systematically executes the instructions in its genome. These instructions determine the organism’s behavior, including its ability to sense and change properties of its environment. Organisms are *self-replicating* meaning that the genome itself must contain the sequence of instructions to create an offspring. When an organism replicates, a target cell for the offspring is selected from the environment at random, and any previous inhabitant of the target cell is replaced (killed and overwritten) by the offspring. In general, each population starts with one or more organisms that are capable only of self-replication, and new genomes are produced through random mutations introduced during replication. Mutation types include substitutions, insertions, and deletions.

2.1.1 Instructions

The Avida instruction set includes commands to perform basic computational tasks (addition, subtraction, and bit-shifts), control execution flow, engage in communication or other environmental interaction, and replicate. The standard Avida instruction set is Turing complete and is designed so that random mutations will always yield a syntactically correct program, albeit one that may not perform any meaningful computation [85]. These instructions are described in detail in [87]. We included additional instructions to enable coordination among organisms. These instructions are summarized in Table 2.1.

Instruction	Description
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 <i>BX</i> 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 <i>BX</i> .
bcast1	Sends a message to all neighboring organisms.
get-cell-xy	Sets register <i>BX</i> and <i>CX</i> to the (x, y) coordinates of the caller.
collect-cell-data	Sets register <i>BX</i> to the value of the cell data where the caller lives.
donate-resources-to-group	Donates the resources the organism has accumulated to the group.

Table 2.1: Coordination instructions for this study. These instructions are in addition to the general Avida instruction set.

2.1.2 Tasks and Resources

Organisms in Avida can perform *tasks*, which enable them to metabolize resources from their environment to gain additional virtual CPU cycles. For many of the division of labor experiments described in this dissertation, we created tasks for each possible role. An organism fulfills a role by performing its associated task. For some of our 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 other experiments, we required the organisms to perform bitwise Boolean logic operations on 32-bit integers. (Lenski *et al.* provide detailed examples of these operations [72].) There are nine possible logic tasks: **not**, **nand**, **and**, **ornot**, **or**, **andnot**, **nor**, **xor**, **equals**.

Tasks consume resources. Generally, metabolizing these resources affects the *merit* of an organism, which determines the rate at which its virtual CPU will execute relative to the other organisms in the population. For example, an organism with a merit of 10 will, on average, execute twice as many instructions as an organism with a merit of 5. Since organisms self-replicate and compete for space, an organism with a higher merit will out-compete other organisms to dominate the population, if all else is equal (execution length, impact of mutations, etc.). However, in some experiments, we configure all organisms to have the same merit and thus run at the same speed. This configuration results in competition primarily occurring at the group level. In these experiments, tasks still consume resources. However, these resources are donated to the group using the **donate-resources-to-group** instruction.

The amount of resources consumed by an organism for performing a task can be either a statically defined quantity (that results from an unlimited resource) or a dynamically changing amount (that results from a limited resource). In most Avida experiments, resources are *unlimited* such that the maximum amount the organism can use is always available. In this case, the amount of merit that an organism gains for completing a task depends on a user-defined constant called the task's *bonus value*. When an organism performs a task, the organism's merit is multiplied by the task's bonus value. For example, if an organism performs a task with a bonus value of 2, its merit is doubled. 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 typically 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. An organism is able to consume a percentage of the resources available (typically 5%). Thus, when many organisms execute a task associated with a limited resource, the amount of resource available decreases until additional organisms would receive a larger amount of resources by completing a task associated with a different resource. Within our experiments, limited resources are used to reward organisms for performing different types of tasks.

2.1.3 Groups

To study group behavior with Avida, we divide the population of digital organisms into distinct subpopulations and link their long-term fates. Groups¹ can comprise a set of genetically-clonal or genetically-heterogeneous organisms. Further detail regarding the differences in behavior will be explained subsequently. As with individual organisms, entire groups replicate and compete for space, but at a much greater time-scale. When a group replicates, another group from the population is selected as its target. The organisms in the target group are all removed, and a subset of the organisms from the source are placed into the target. Figures 2.2 and 2.3 depict a heterogeneous and clonal group replicating, respectively. In both cases, group A (or a derivative thereof) replaces group B.

In these studies, we make use of two different forms of group competition. Both types of group competition result in group replication. First, for some experiments, we use *tournament selection* in which groups compete every 100 updates² based on a fitness function, where a group's success is determined by the behavior of its constituent organisms. Each tournament contains a set of groups selected at random, and the group with greatest fitness

¹Within other Avida papers, we will sometimes refer to these groups as demes or colonies.

²An *update* is the unit of experimental time in Avida corresponding to approximately 30 instructions per organism.

is replicated (ties are broken randomly). Second, some of our experiments use *task-based group competition*, where a group replication event is triggered by the collective behavior of individuals within a group. For example, a group could be set to replicate once its constituent organisms have consumed a certain amount of resources.

2.1.4 Genetically Heterogeneous Groups

Some of our studies investigate the effects of *multilevel selection*, where the group-level (i.e., colony-level) pressure [111] selects for division of labor and the individual-level pressure rewards different roles unequally. The theory of multilevel selection, originally called *group selection* was proposed in 1962 by the biologist V. C. Wynne-Edwards [132], and later formalized into multilevel selection theory by D. S. Wilson and E. Sober [112, 130]. At a high level, multilevel selection theory states that groups of individuals may be vehicles for selection, similar to how a single individual is a vehicle for the selection of its components. In other words, multilevel selection posits that the survival of the individual is *linked* to the survival of the group. 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 eusocial insects, such as ant and bee colonies [9, 23, 53, 97, 105, 111].

To implement multilevel selection, when a group is replaced during replication, all organisms from the source group are copied into the target group, overwriting its previous inhabitants. We refer to this replication method as *full group replication*. Within each group, organisms are still able to self-replicate. This setup enables groups to potentially maintain *genetic heterogeneity*, since all organisms are copied when the group replicates.

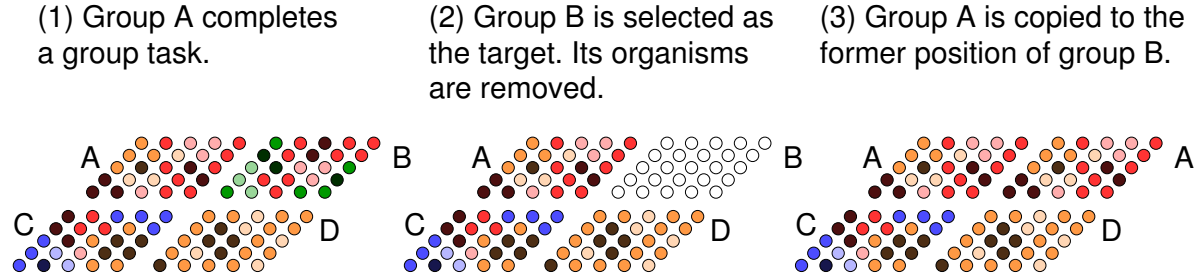


Figure 2.2: Heterogeneous Group Replication Process: (1) An Avida population containing four heterogeneous groups. Group A completes a group task or wins a group competition. (2) At this point, group B is selected to be replaced and all its organisms are removed. (3) A perfect copy of group A is created and is loaded into group B.

Thus, an individual’s survival is dependent not only on its ability to out-compete its neighbors for the limited space available in its group, but also on the collective behavior of the group. Figure 2.2 depicts a world comprising four heterogeneous groups (A, B, C, and D). During a replication event, group A replaces B and thus the overall population contains two identical copies of group A.

2.1.5 Genetically Clonal Groups

Some of our other studies examine the evolution of genetically homogeneous groups. To enable these *clonal* groups, we attached a *germline*, or heritable lineage, to each group in the Avida population [65]. In most multicellular organisms, genetic material is transferred from parent to offspring along the germline [24]. *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 the organism itself. This reproductive division of labor is similar to that exhibited by eusocial colonies where only the queen’s genetic material affects subsequent colonies. Within our studies of clonal groups, we do not introduce mutations during the process of individuals replicating within a group. Thus, except under specific circumstances where an individual intentionally creates an offspring that is different than itself, all organisms have identical

genomes. Figure 2.3 illustrates the *germline replication* process: (1) a colony completes a group task; (2) another colony from the population is selected as the target of the replication and the organisms within the target colony are removed; (3) the germline from the source colony is used to produce the germline of the new colony possibly with mutations; (4) one individual from the new germline is placed into the target colony; (5) the original colony is reset to one organism.

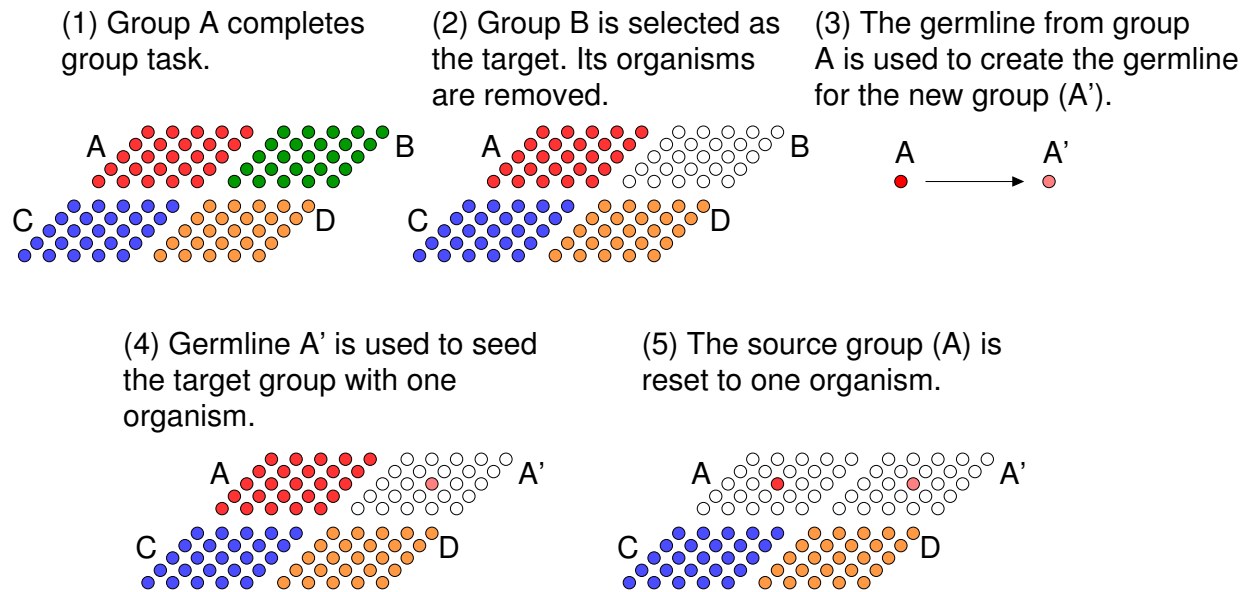


Figure 2.3: Clonal Group Replication Process: (1) An Avida population containing four clonal groups. Group A completes a group task or wins a group competition. (2) At this point, group B is selected to be replaced and all its organisms are removed. (3) The germline of group A is used to create the germline of the new colony. (4) One individual from the new germline is placed into the target group. (5) The source group A is reset to its original state with one individual.

These studies of genetically clonal groups differ from the genetically heterogeneous studies in three key ways: First, for clonal groups, mutations occur along the germline, rather than when an individual organism replicates within a group. Second, for clonal groups, the target group is seeded with one organism from the germline, rather than copies of all organisms in the group when it replicates. Third, because the organisms within a group are identical, for clonal studies, we applied only the group-level component of multilevel selection pressure and

thus do not apply individual task pressures. Specifically, we set the merit of all individuals to be the same thus ensuring they all run at the same speed.

2.1.6 Experimental Configuration

For each treatment, in all of the studies, we ran 20-50 replicate runs to account for the stochastic nature of the evolutionary process. When we used groups, the *default group topology* is 400 groups that each contained up to 25 organisms on a 5×5 toroidal grid. Further experimental details will be provided for each experiment.

2.2 Measuring division of labor

To measure the amount of division of labor present within a colony, we use Shannon mutual information as proposed by Gorelick *et al.* [48]. The formula for Shannon mutual information is:

$$I = \sum_{i=1, j=1}^{m, n} p_{ij} \ln\left(\frac{p_{ij}}{p_i p_j}\right) \quad (2.1)$$

where p_i is the probability of an individual, p_j is the probability of a task, and p_{ij} is the joint probability of a given individual i performing task j . For this study, we treat the probability of all individuals (p_i) as equal. The probability of a task (p_j) is given by the number of times that type of task was performed divided by the number of individuals.

Intuitively, Shannon mutual information captures two reciprocal pieces of information: (1) given an individual, how much information do we have about the type task it performs, and (2) given a type of task, how much information do we have about the individual that performed it. Figure 2.4 depicts how Shannon mutual information varies with different relationships among individuals and tasks. If all the individuals specialize on performing

the same task, then Shannon mutual information (and division of labor) are low (e.g., Figure 2.4(a)). If an individual performs many different types of tasks (e.g., a generalist strategy), then Shannon mutual information will also be low (e.g., Figure 2.4(b)). In contrast, information will be high when an individual performs only one type task and when a type task is performed by only one individual (Figure 2.4(c)).

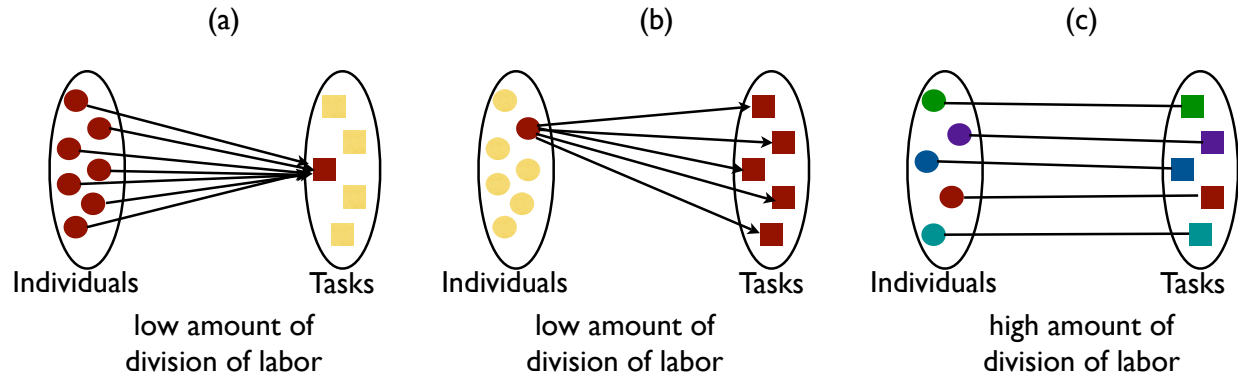


Figure 2.4: A depiction of the amount of Shannon mutual information present among a group of tasks and a group of individuals. (a) All individuals specialize on performing the same task resulting in a low amount of division of labor. (b) A generalist individual performs all of the tasks resulting in a low amount of division of labor. (c) Each individual specializes on performing a different task resulting in a high amount of division of labor.

Chapter 3

Direct Selection for Division of Labor

In this chapter, we explore whether game theory and Avida can be used to evolve groups of organisms that exhibit division of labor when we directly select for it. For these studies, we start with the simplest possible form of division of labor where there are two possible roles (leader and follower). From there, we expand the number of roles and the complexity of roles. We then study whether division of labor will evolve, despite these complications, in genetically heterogeneous and clonal groups.

3.1 Leaders and Followers:

Frequency-Dependent Selection

In nature, division of labor enables a group of collaborating individuals to collectively achieve better outcomes than if each individual acted as a generalist in isolation [25, 51, 113]. Such labor specialization can produce greater net benefits for the group as a whole, and may be a key factor in economic growth and stability [32, 108]. However, observations of collective action in humans and other animals reveal patterns of resource inequity, where individuals in certain labor roles appear to end up with more benefits than others, which may have

destabilizing effects for groups and societies [17, 91]. For example, leadership is a common role found within multiple species, where the benefits of leadership are significantly greater than that of a follower. In human societies, the salary of leaders of organizations commonly is many times greater than the salary of the average worker [125]. Typically, those in dominant roles are seen to reap greater benefits than those in subordinate roles in many dyadic or group interactions [19, 52]. Thus, although group members can achieve better outcomes if the group successfully coordinates labor specialists, each individual is also under pressure to perform the labor role within the group that yields the greatest individual benefit. These inequities result in the question: if it is the case that dominant leader roles in group interactions pay so much better than subordinate follower roles, then why would anyone select to perform a follower position?

From a game-theoretic perspective, resource inequity raises important questions relevant to the understanding of the evolution of division of labor strategies, and how coordination and collective action problems are interrelated. Specifically, individuals must coordinate their labor roles in order to achieve the collective action objective when separate roles may yield short-term differences in payoffs. In this study, we propose a game-theoretic model that examines division of labor within dyads, where there are two roles: “risky” leaders and “safe” followers — each role with different payout yields after each interaction. The general idea behind our model is reminiscent of a simple “hawk/dove” game concept, but is adapted to simulate an intuition about how costs, benefits, stochastic and risk-management processes that affect “real world” leader-follower dynamics might be relevant to the outcomes.

Our model begins with a familiar intuition about social groups: there two types of agents in interacting dyads. The first type of agent initiates an interaction and assumes greater risk (the *leader*), but receives a larger share of the collective action prize. Likewise, the agent adopting a “you first” strategy (the *follower*) takes less risk and receives a smaller share of the collective action prize. As is typically assumed to be the case in most economic transactions,

both agents in a dyad receive a division of labor benefit if they successfully coordinate an interaction where one individual is a leader and the other is a follower. Dyads of two leaders or two followers do not receive a benefit. We specify this model analytically and then confirm the results using computer simulations. Our model reveals that the proportion of labor roles can reach a stable equilibrium, where each role has a similar average net payoff over the long run, even when individuals playing a certain strategy (e.g., “follower”) always have a lower payoff than the other strategy (e.g., “leader”) in any given interaction that contains one of each. In the analysis of our model that yields this result, we shed light on the questions of (a) why leaders are less numerous than followers, and (b) why some people may choose lower paying follower roles rather than “higher-paying” leadership roles. In doing so, we also explore the notion of how leader-follower interdependent strategies that can give rise to the emergence of leaders who incur greater costs than followers in transactions, yet are still maintained in a population because of frequency-dependent selection. Specifically, we demonstrate that by varying the relationship among the benefits and costs of coordination for the two roles, we are able to produce populations that exhibit these characteristics.

Our model is designed to reflect the tradeoffs, risk, and chance events characteristic of leaders and followers in the real world. Specifically, groups can benefit from transactions involving division of labor, but the costs and benefits may be distributed unevenly. Since moving first to initiate a transaction is often riskier than receiving an overture to interact, the individual with the riskier strategy should expect to receive a larger fraction of the group benefit. For example, in a hunting party, a leader may charge the prey into a trap set by the follower. If the prey mounts a defensive attack, then the leader is likely to bear the brunt of the counter-attack, and is thus incurring greater risk of injury or death. In order to offset the cost of this increased risk, the leader may also receive a disproportionately large share of the meat. This situation is roughly analogous to the risk-management strategies behind “hazard pay” in employment economics. Crucial to the outcomes may be the consideration by each

player as to how much one stands to gain from dyadic transactions, how much one can lose, the probability of failure (e.g., likelihood of death), and whether information transmission errors are common. These may be important parameters in any meaningful description of the evolution and dynamics of leadership and followership. In the following section, we first describe our leader-follower model in “top down” fashion, and then show empirical results from computer simulations in order to illustrate the effects of modifying key parameters within the model.

3.1.1 Methods

Our model assumes two possible roles: leader and follower with the former incurring greater risk of failure than the latter. This added risk comes in the form of a *leadership cost* that solely afflicts the player in the leader role, but which can be offset by the coordination benefits that both players receive, and can be further offset by a multiplicative benefit for taking the leader role. Table 3.1 depicts our payoff structure for a dyad. The payoffs for each interaction are determined by: c , the cost of taking the leadership role (which can be potentially large, as in the case when one is killed, but may translate into a small cost if the risk is marginal); b , the benefit for a successful division of labor transaction (shared equally between players); and a , the leadership bonus of being a leader when a successful transaction is coordinated (indicated in our model as a multiple of the shared benefit).

In a dyadic transaction, if both individuals opt for the leadership role, then they both incur cost c without benefits a or b (e.g., as if both charge the animal and thus incur the risk of being gored, but neither set the trap to capture the prey). If both opt to be followers, then they incur no cost (e.g., there is no risk of injury or death by not charging the prey), but neither do they earn a benefit (e.g., there is no fitness benefit to gaze at peacefully grazing prey animals while starving). If one is a leader and the other is a follower, then the dyad has successfully coordinated and a successful division of labor transaction occurs (e.g., they

successfully hunt). In these successful transactions, followers and leaders each earn a division of labor benefit (b), but the leader also is awarded a leadership bonus (a) net a leadership cost (c). Thus, in successful transactions, leaders may receive greater rewards than followers, but at a price.

	Leader	Follower
Leader	$-c, -c$	$b * a - c, b$
Follower	$b, b * a - c$	$0, 0$

Table 3.1: The payoff matrix for the leader-follower game, where a is the multiplicative benefit of leadership, b is the coordination benefit, and c is the cost of leadership. In general, leaders have higher payoffs if the dyad coordinates roles, but a lower payoff if they do not.

Formal Model

The formal model describes how a sufficiently large population will behave at equilibrium. The composition of the population at equilibrium is determined by the values selected for the leadership cost, multiplicative benefit of leadership, and division of labor benefit. We derive the frequency of leaders at equilibrium, p , from the payoff matrix. Let $w(L)$ and $w(F)$ be the payoff of leaders and followers, respectively:

$$w(L) = -cp + (1 - p)(ba - c) \quad (3.1)$$

$$w(F) = bp \quad (3.2)$$

The payoff of leaders ($w(L)$) is derived by summing the payoff of leader-leader dyads multiplied by the frequency of this dyad composition occurrence within the population and the payoff of leader-follower dyads multiplied by the frequency of this dyad composition occurrence within the population. Specifically, the first term is the payoff of leader-leader dyads multiplied by the probability of this occurrence, which is the frequency of leaders in the population (p); provided cost is greater than 0, this term is negative. The second term is

the payoff of leader-follower dyads multiplied by the frequency of followers in the population $(1-p)$.

For the payoff of followers $(w(F))$, the sole term is the probability of a leader-follower dyad multiplied by the division of labor benefit. (Recall that a follower-follower dyad neither incurs cost nor accrues a coordination benefit.)

At equilibrium, the benefits of leadership and followership are equal. Using this observation, we derive an equation representing the frequency of leadership using basic substitution as follows:

$$w(F) = w(L) \tag{3.3}$$

$$bp = -cp + (1 - p)(ba - c) \tag{3.4}$$

$$bp = -cp + ba - bpa + cp - c \tag{3.5}$$

$$p^* = (ba - c)/[b(1 + a)] \tag{3.6}$$

The last expression (Equation 3.6) describes how the frequency of leaders in the population (p^*) is expected to vary as a function of net leader benefit (numerator) scaled to the aggregate benefit of the leaders and followers combined (denominator).

Simulation

To better understand the temporal behavior of populations as they reach equilibrium, we developed a simulation, which is available online at <https://www.msu.edu/~hjpg/Leader-Follower-PD-EA.html>. Figure 3.1 depicts a screenshot of the game. Prior to the start of the game, the parameters of simulation are entered. Similar to the formal model, configuring the parameters includes specifying the leadership cost, division of labor benefit, and leadership benefit. Within the simulation, the average leadership cost from the formal model is divided into two elements: a per interaction probability of a leader incurring the

leadership cost (**probability of leadership cost**) and the leadership cost (**leadership cost**). While these parameters would not affect a sufficiently large population at equilibrium, they may affect smaller populations and/or the temporal behavior of populations. In addition to the parameters present in the formal model, the population size and initial population composition can also be specified. The initial population composition determines the number of individuals that will start as either leaders and followers.

Each of the simulations presented begin with a population size of 1,000, where all members of the population are initially followers. We run the simulation for 100 generations, where each generation comprises 5,000 games. (We selected 5,000 so that, on average, each individual participates in 10 games per generation.) In a given game, two individuals are selected from the population at random to form a dyad and play the leader-follower game. Their payoffs are determined using the payoff matrices. An individual's net payoff for a generation is the sum of the payoffs received in each game it has participated in. An individual's strategy remains constant during a generation. At the end of a generation, the next generation of individuals is selected, where an individual that has a higher net payoff is more likely to be selected to produce individuals in the next generation. Specifically, we use fitness proportional selection (roulette-wheel selection) [4] to select individuals from the current generation to produce individuals in the next generation. As a result of this selection process, a strategy that has yielded higher net payoffs is more likely to proliferate. Conversely, a strategy that has yielded lower net payoffs is less likely to proliferate. Additionally, 1% of the population mutates by switching strategies. For example, if a leader mutates, then it becomes a follower. Mutation prevents a strategy from going extinct. The results of the simulation are presented in terms of the population composition (the percentage of the population that are leaders and followers) over time. While the formal model describes the ideal behavior of an infinite population at equilibrium, the simulation describes the behavior of a finite population playing a limited number of games over time. These simulation

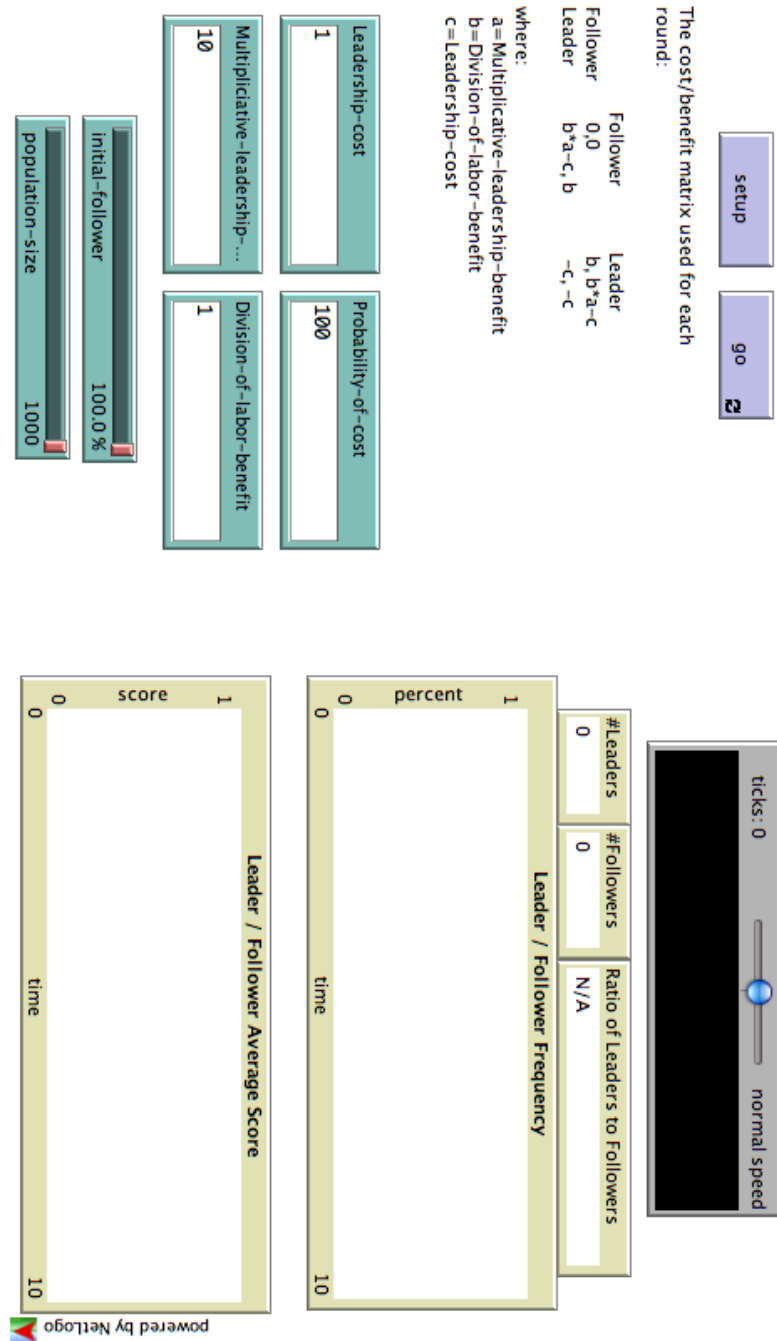


Figure 3.1: A screenshot of the leader-follower game developed using NetLogo and available online. The game enables the user to configure the costs of leadership, benefits of leadership, and the benefits of division of labor.

constraints in conjunction with randomness present within the simulation may result in the simulation results deviating from the formal model. However, these deviations are useful for understanding the behavior of human populations, which are subject to many of the same constraints.

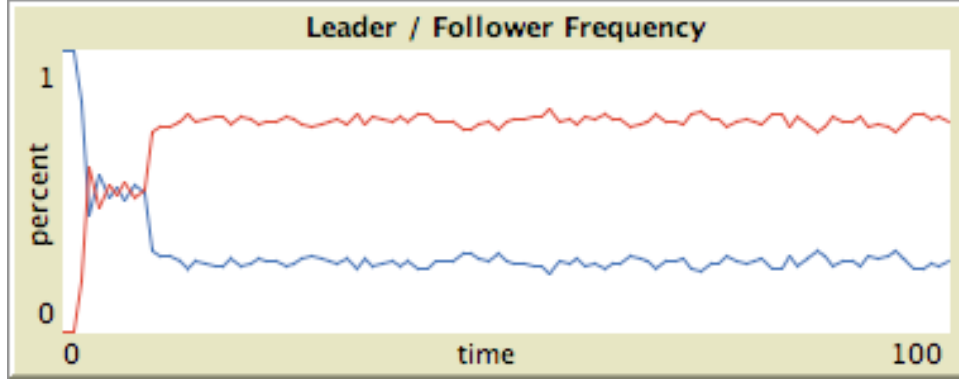
3.1.2 Results

In this section, we demonstrate that the leader-follower game-theoretic model produces frequency-dependent selection that can result in populations of various compositions of leaders and followers, including compositions in which leaders greatly outnumber followers.

We use the formal analytic model to identify cost benefit combinations that result in populations where the vast majority of individuals ($> 75\%$) are leaders. When there is no cost for leadership ($c = 0$), then a leadership benefit (a) of greater than or equal to three times the division of labor benefit (b) produces a population that is primarily leaders. However, even adding a slight cost of 1 causes the benefit of leadership (a) to need to rise to 7 times the division of labor benefit (b) to produce a primarily leader population. Costs of greater than 2 do not reach this population composition even with leadership benefits of up to 10-fold.

Figure 3.2(a) depicts the simulation results for one configuration in which leaders outnumber followers. Here, we set the benefit of division of labor (b) to 1, the benefit of leadership (a) to 3, and cost (c) to 0. Although, at equilibrium, the net payoff for both strategies is equal, for a dyad that exhibits division of labor, the leader payoff is 3 and the follower payoff is 1. The simulation data show the population converging on a 75% leader population composition. The stochastic nature of the dyad pairings, roulette-wheel selection, and mutation cause the simulation data to vary slightly from the formal model. Specifically, the simulation takes a period of time to reach equilibrium, and, even once equilibrium is reached, experiences some minimal fluctuations.

(a) Leaders greatly outnumber followers.



(b) The number of followers and leaders is almost equal.

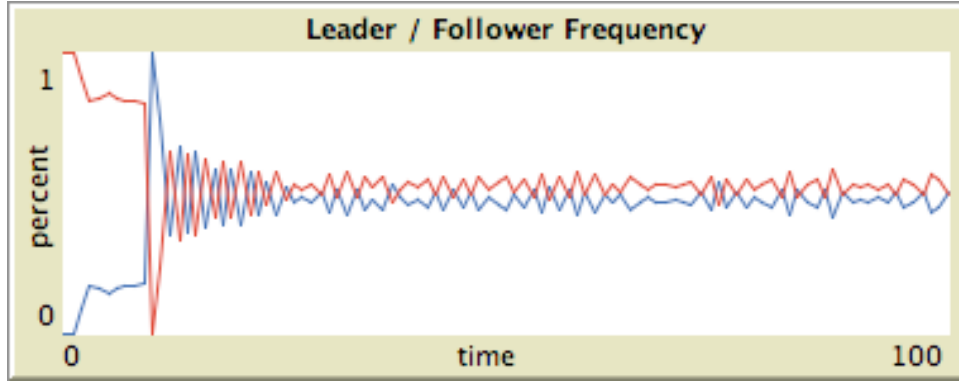


Figure 3.2: The game-theoretic simulation results for various combinations of costs and benefits. The red line represents the percentage of the population that selected the leader strategy and the blue line represents the percentage of the population that selected the follower strategy. In the first treatment (a), leadership benefit is 3, coordination benefit is 1, and leadership cost is 0. These values result in many leaders. In the second treatment (b), leadership benefit is 5, coordination benefit is 1, and leadership cost is 2. These values result in an equal number of leaders and followers. These results demonstrate that frequency-dependent selection can be used to produce various population compositions of leaders and followers even if the payoffs per interaction are not equal.

There are several possible configurations that result in the frequency of leaders and followers being approximately equal. When cost (c) is equal to 0 and the benefits of division of labor (b) and benefits of leadership (a) are 1, then the population composition is evenly split. With higher costs, increasingly large leadership benefits (a) are required to achieve this population split. For example, when cost is equal to 2, then a leadership benefit of 5 is

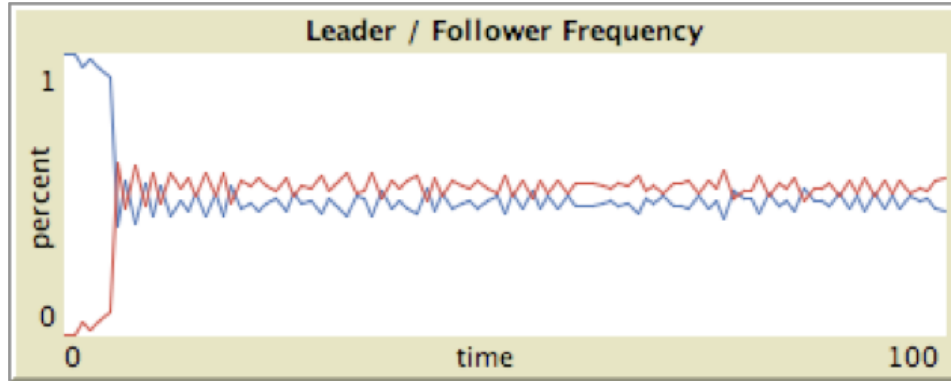
required. For these values, if a dyad exhibits division of labor, then the leader has a payoff of 3 and a follower has a payoff of 1. However, when the dyad comprises two leaders, then both have a payoff of -2.

Figure 3.2(b) depicts simulation results for one such configuration. Here, we set the benefit of division of labor (b) to 1, the benefit of leadership (a) to 5, and cost (c) to 2. Additionally, in this case, we also varied the initial population composition. In general, we start with a population comprised entirely of followers. In this case, the simulation started with a population entirely of leaders. Here, the starting configuration affected the time necessary to converge, but did not affect the equilibrium population composition.

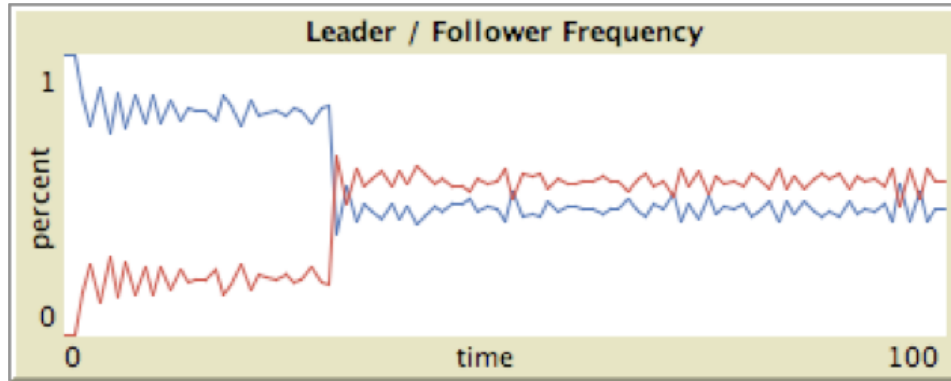
Next, we examine how the results of the simulation, which uses a finite population and probabilistic cost, may differ from the formal model. We examine the role of probabilistic cost by keeping the same mean cost ($\text{cost} * \text{probability}$), but varying the specific probability and cost values. Specifically, we used the same leaders parameter values (Figure 3.2 (b)), where the mean cost is 2, the division of labor benefit is 1 and the leadership benefit is 5, as a baseline. Thus, for the probabilistic cost treatments we used: (a) a cost of 2.67 with a probability of 75%, (b) a cost of 4 with a probability of 50%, and (c) a cost of 8 with 25%.

Figure 3.3 depicts the results of the probabilistic cost treatments. In general, the results of the simulation agree with the formal model and there is a balance of leaders and followers. However, as the probability of the cost is reduced, the volatility of the results increases. For example, in Figure 3.3(c), the population temporarily fixes on leaders, and then followers, before settling in around equilibrium. Also, as the probability is reduced, the number of leaders increases slightly. This results from the dynamics of the simulation in a fixed population and with randomness.

- (a) Few leaders: division of labor benefit = 1; leadership benefit = 5; cost = 2.22; probability of cost = 75%



- (b) Few leaders: division of labor benefit = 1; leadership benefit = 5; cost = 4; probability of cost = 50%



- (c) Few leaders: division of labor benefit = 1; leadership benefit = 5; cost = 8; probability of cost = 25%

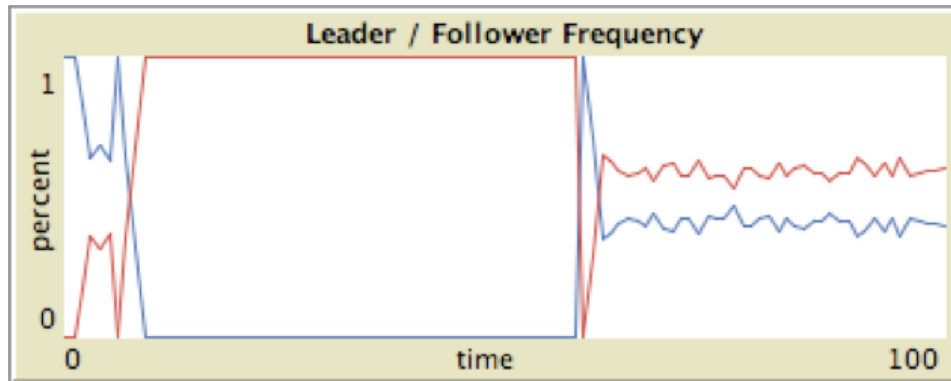


Figure 3.3: The simulation results for various combinations of probabilistic costs given the same division of labor benefit, leadership benefit, and mean cost. The red line represents the percentage of the population that selected the leader strategy and the blue line represents the percentage of the population that selected the follower strategy.

3.1.3 Discussion

Our analysis of the interdependent roles of leaders and followers in coordinated action yields the implication that leaders who bear a cost can emerge, even if their net payoff in each coordinated transaction is lower than that for followers. Similarly, followers who bear a cost can also emerge.

In the same way that a leader glut reduces the coordination benefits of leadership, a follower glut poses the same problem for followers. The key idea, is that there are benefits to encountering other like-types with complementary division-of-labor strategies. Thus the payoffs can be lower for leaders than for followers on coordinated transactions, yet leaders can still evolve if they are rare enough to exploit the follower glut. As such, leaders can make up for lower net payoffs on each outcome by having a greater number of payoffs than followers. This situation is possible if (1) the benefit exceeds zero, and (2) leaders are more rare than followers. Thus, situations where there are more followers than leaders can evolve, and vice-versa. Since in human social contexts, the former more commonly emerges, it is possible that the net benefits of leadership in most human groups may actually be very low. The widely touted statistic is that Fortune 500 CEOs are compensated at 200 times the average workers (20,000% higher) is likely to be an expression of the extreme tail end of a distribution of a horde of others struggling to keep stride or who have met complete ruin.

Potential extensions of this work could include using coordination models to address a range of problems that require collective action where one agent is at greater risk of injury or failure, and so is compensated at some greater rate than are others. Other examples might include defensive or aggressive attacks against a predator, criminals, or out-group combatants. For a richer account, leader characteristics related to trait-related assortment dynamics (where individuals actively seek out complementary types) or self-appraised abilities to be a better leader or follower are necessary. The basic enterprise may be extended to group action problems such as those involving venture capital; income disparity; intergroup conflict and

aggression; and patriotism and warfare.

3.2 Leaders and Followers: Multilevel Selection

A second possible explanation for why leaders and follower coexist within groups is multilevel selection [125]. In this section, we use Avida to explore whether multilevel selection could be used to evolve a genetically heterogeneous group that has one leader and many followers [43, 45]. Similar to our game-theoretic work, this setup is designed to reflect a leader-follower situation, where it is desirable for the group to have one leader, and yet the rewards for the leader may be significantly different than those of a follower. This hypothesis serves as an alternative explanation to pure frequency-dependent selection.

3.2.1 Methods

In this set of experiments, we investigate whether groups of genetically-heterogeneous organisms can evolve to coordinate roles, even when these roles have unequal rewards. The different rewards associated with roles provides a pressure for individuals to specialize on the most rewarded role, even if this behavior is detrimental to the performance of the group. To test a group’s ability to support less-rewarded roles, we designed a leader-followers situation, where it is desirable for the group to have one leader, and yet the rewards for the leader may be significantly different than those of the followers. At the individual level, we rewarded organisms who set their role-id register to 1 (the leader role), while no rewards were granted for selecting other role-ids. We then specified a group-level pressure to limit the leader role to only a single organism. Specifically, the group fitness function that we used here was:

$$F = \begin{cases} 1 + n & \text{if } n < 25 \\ (1 + n - (o - d)^2) & \text{if } n = 25 \end{cases} \quad (3.7)$$

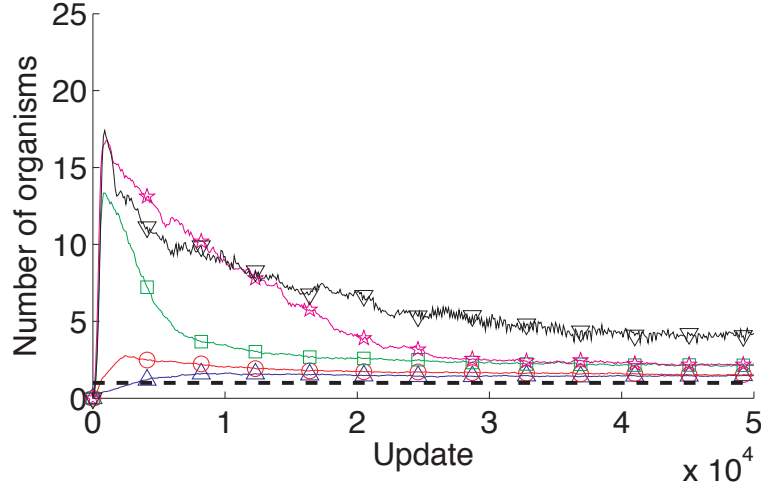
where F is the fitness of a group, n is the number of organisms that have set any role-id, d is the target number of organisms that should perform the leader role (1 in this case), and o is the actual number of organisms that perform the leader role. Thus, this fitness function rewards groups whose constituent organisms set their role-id register, and then further rewards for coordinating roles such that only a single organism assumes the role of leader.

For these experiments, we used the default group topology with genetically-heterogeneous organisms. These groups competed every 100 updates in tournament selection; fit groups propagate using full group replication. We ran 30 trials of each experiment.

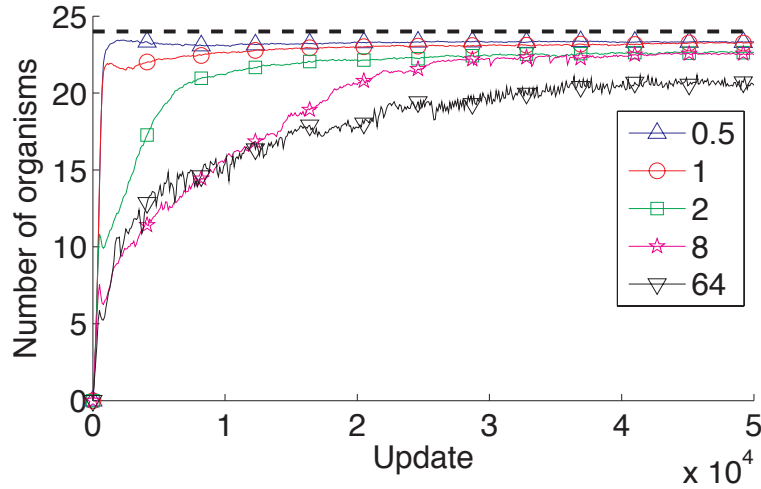
3.2.2 Results

Figure 3.4 depicts the results of varying the multiplicative benefit of the leader role from 0.5 (a penalty) to 64 (a large reward) across 30 trials. Specifically, Figure 3.4(a) depicts the mean number of leaders for the different trials and Figure 3.4(b) depicts the mean number of followers. The dashed lines in both plots denote the ideal configuration – one leader and 24 followers. In general, by 50,000 updates, the average group had reached an equilibrium between leaders and followers, where the number of leaders was fewer than five in all treatments. Unsurprisingly, because the group was assigned a fitness of 0 if it did not have a leader, the groups maintained more than one leader for redundancy. The larger the benefit of leadership, the slower the population was to reach equilibrium and the larger the number of leaders. These results demonstrate that Avida is capable of evolving groups that exhibit leader-follower division of labor even when the rewards are greatly disparate.

We next performed several additional experiments that varied key parameters to better understand the factors underpinning division of labor in this environment. First, to assess the generality of these results, we ran leader-follower experiments where we set the desired number of leaders to be 13 (approximately half of the group) using the group fitness function



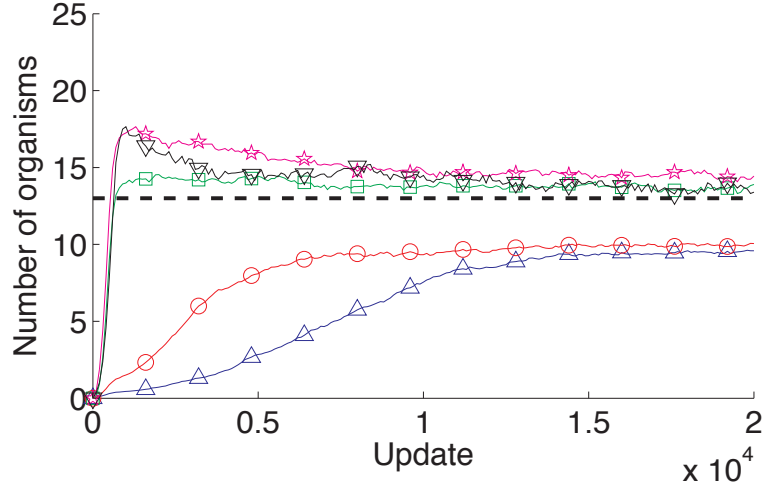
(a) Number of leaders



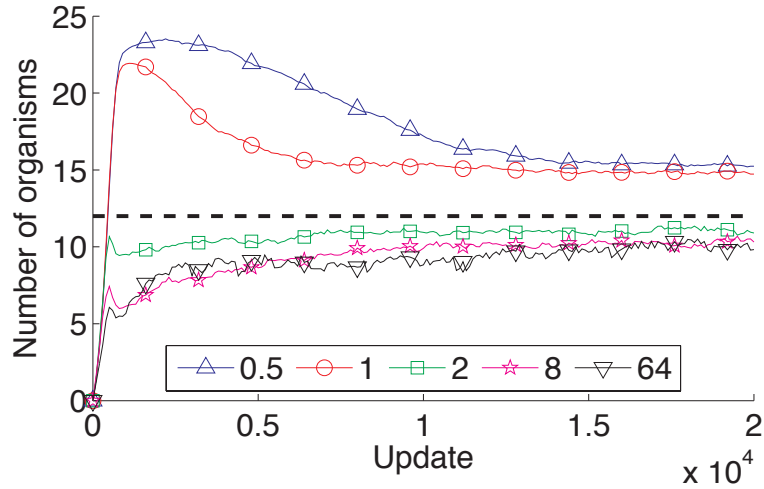
(b) Number of followers

Figure 3.4: The mean number of leaders (a) and followers (b) when each group’s target is one leader and 24 followers. The dashed line represents the target value in each case. Each line represents a different treatment that has a different multiplicative bonus value of leadership varying from 0.5 to 64. Overall, the group-level component of the multilevel selection pressure is sufficient to produce groups with few leaders and many followers, despite the appreciable pressure for individuals to be leaders.

described by Equation 3.7. Figure 3.5 depicts the results of this experiment. We see that the population reached equilibrium by 15,000 updates and was nearly evenly divided between leaders and followers. Similar to the previous experiments, as the reward for leadership increased, so did the number of leaders.



(a) Number of leaders



(b) Number of followers

Figure 3.5: The mean number of leaders (a) and followers (b) when each group's target is 13 leaders and 12 followers. The dashed line represents the target value in each case. Each line represents a different treatment that has a different multiplicative bonus value of leadership varying from 0.5 to 64. Overall, the group-level component of the multilevel selection pressure is sufficient to produce groups with few leaders and many followers, despite the appreciable pressure for individuals to be leaders.

3.2.3 Discussion

Division of labor when roles are rewarded unequally can be recast into a discussion of altruism. *Altruism* occurs when an individual sacrifices fitness for the benefit of another

individual. Altruistic acts are further divided into those that are *strongly altruistic*, where the individual sacrifices all of its fitness, and those that are *weakly altruistic*, where an individual sacrifices fitness relative to the rest of the group, but altruistic groups out-perform non-altruistic groups [129]. In this study, we used multilevel selection to evolve groups that exhibit division of labor within a leader-follower system, where an organism that selects a less-rewarded task is performing a weakly altruistic act – the organism receives less fitness than others in its group that perform better rewarded tasks, but the group, and thus its altruists, out-competes non-altruistic groups. It is often the case that explanations from a multilevel selection perspective can also be interpreted from a kin selection perspective, and both are equally valid yet provide different intuitions [30]. For this study, interpreting the results from a multilevel selection perspective provides the most intuitive explanations.

Numerous evolutionary computation approaches have been used to study the behavior of cooperative groups comprising heterogeneous members [35, 90, 94, 126, 134]. Two key differentiating characteristics for these approaches are the *level of selection* used (i.e., individual or group) and whether or not *division of labor* occurs. Ecological approaches [35] use individual-level selection in concert with limited resources to promote the evolution of specialists. Some coevolutionary approaches [94, 134] evolve cooperative groups using individual selection, where different species are isolated in distinct subpopulations. Cooperation among these species occurs only at the time of fitness evaluation, when individuals from one species are evaluated with representatives from each of the other species. Perez-Urbe *et al.* [90] and Waibel *et al.* [126] overview work performed in this area, and also describe the effects of varying the levels of selection and population composition (i.e., heterogeneous or homogeneous populations) [126]. However, these prior studies do not address multilevel selection, where organisms experience individual-level competition to survive and also group-level pressure to cooperate.

In this preliminary study, we investigated whether leaders and followers, which represent

roles with different rewards, could be maintained within a group through application of multilevel selection. We found that the group-level fitness component was a sufficient pressure to maintain both roles. Next, we expand upon this result to capture more complex division of labor arrangements in which organisms can select from more than two roles.

3.3 Division of Labor: Heterogeneous Groups

In this section, we extend our leader-follower approach to evolve more complex forms of division of labor that include both additional roles for the organisms to perform and more challenging tasks associated with the roles. For these studies, we directly reward for division of labor and observe whether groups are able to evolve to coordinate roles and, if so, what mechanisms they use to do so. In the following, we describe our experiments that examined whether multilevel selection can be used to evolve groups of heterogeneous organisms that exhibit division of labor [43, 45].

3.3.1 Methods

For these experiments, we used the default group topology (i.e., the Avida world comprises 400 groups) with genetically-heterogeneous organisms. These groups compete every 100 updates in a tournament, where fit groups reproduce using full group replication. Organisms perform division of labor roles by completing tasks. The nature of these tasks varies among experiments. We ran 30 trials of each experiment.

3.3.2 Results

For this study, we organized our experiments to address a series of increasingly difficult challenges. First, we tested the coordination capabilities of the organisms by assessing how many roles they were capable of performing. Second, rather than having an organism select

a role, we conducted experiments in which a role was associated with a labor task that the organism had to perform. Third, we analyzed the groups that successfully performed all of the labor roles to determine their strategy.

Can groups evolve to coordinate roles?

We designed our first experiment to test the most basic question: If the benefits associated with each task are equal, then can groups of organisms evolve to coordinate their roles? If groups of organisms cannot maintain role diversity for this simple group task, then more complex forms of division of labor within heterogeneous groups are most likely not possible. However, if groups of organisms can perform this task, then we have a foundation to pose more complex questions regarding challenging roles with unequal rewards.

To test this question, we used a set of individual-level tasks that reward organisms for assuming roles. Specifically, we considered an organism to have assumed a given role if it sets its role-id register to an element of R_n , the set of valid role identifiers. We define R_n as the set of consecutive integers $\{1, \dots, n\}$; for example, $R_5 = \{1, 2, 3, 4, 5\}$. We consider any organism that sets its role-id register to a value in R_n and then self-replicates as having completed a task; thus, we double its merit. If an organism sets its role-id to a value that is not in R_n and then self-replicates, its merit remains unchanged. Prior to self-replication, organisms may set their role-id registers to different numbers. Only the value of the role-id register at the time of replication affects merit.

Additionally, we impose a group-level pressure for both the number and the diversity of the role-ids. Specifically, we define the group fitness function as:

$$F_{group} = \begin{cases} 1 + n & \text{if } n < 25 \\ 1 + n + r & \text{if } n = 25 \end{cases} \quad (3.8)$$

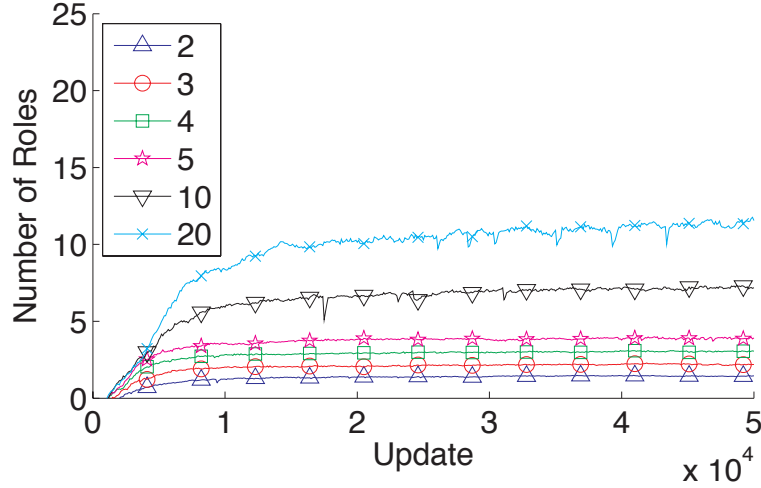
where F_{group} is the fitness of a group, n is the number of organisms that have set any role-

id, and r is the number of unique rewarded role-ids set by organisms in the group. This fitness function includes n to encourage organisms to select roles, even if they are not fully coordinated in their approach. Thus, a group whose constituent organisms do not select any role will be out-competed by a group whose organisms all select the same role, which will in turn be out-competed by a group whose organisms all select different roles. Group fitness was evaluated during the selection tournaments that occurred every 100 updates.

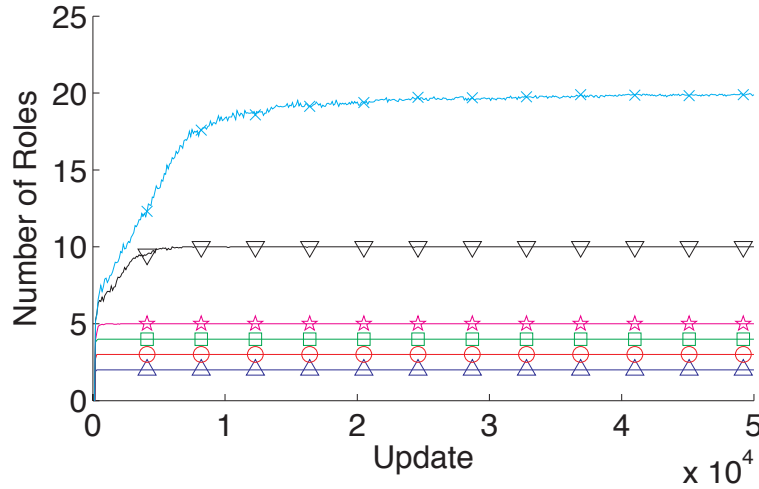
Figure 3.6 depicts the grand mean (mean of mean) and mean of maximum number of unique roles¹ within each group across 30 trials for different values of R_n ranging from R_2 to R_{20} . Each curve represents a different R_n . In general, the highest-performing group from each run achieves the desired number of roles within 5,000 to 15,000 updates. An analysis of the behavior of the groups indicates that they exploited location awareness (using the `get-cell-xy` instruction) to determine which role to perform. Specifically, an organism’s x and y cell coordinates within its environment were used to calculate its role. In general, we found the number of roles performed across treatments to be statistically significantly different from each other at the $p < 0.05$ level (Kruskal-Wallis multiple comparison). However, “near” treatment pairs (e.g., 2-3, 3-4, etc.) were only found to be statistically significantly different from each other when compared independently (Wilcoxon rank sum test, $p < 0.001$ for all near-pairs). Thus, we conclude that multilevel selection is capable of producing division of labor among equally rewarded roles.

To verify that individual selection pressures alone would not explain this division of labor, we performed a control experiment in which we removed the group-level selection pressure. Specifically, we maintained the individual-level reward for selecting a role, however the winner of each group tournament was selected randomly without regard to the diversity of the group (i.e., F_{group} from Equation 3.8 was set to 1 in all cases). Figure 3.7 depicts the mean and maximum number of unique roles within each group across 30 trials for different

¹In subsequent plots, we refer to these as the mean and maximum number of roles.



(a) Grand mean number of roles performed



(b) Mean of max number of roles performed

Figure 3.6: The results of using multilevel selection to evolve groups that comprise varying numbers of roles, where each line represents a different treatment with a different desired number of roles ranging from 2 to 20. The grand mean (a) and mean of the maximum number of unique roles (b) over all 30 trials. In general, the groups evolved to perform the desired number of roles.

R_n under this control treatment. As can be seen here, the mean number of roles for all treatments is zero (Figure 3.7(a)), indicating that the groups did not satisfy the first part of the fitness function, where we require that all organisms within a group select a role. For 2-4 roles, the maximum performance of the 30 groups is the desired number of roles

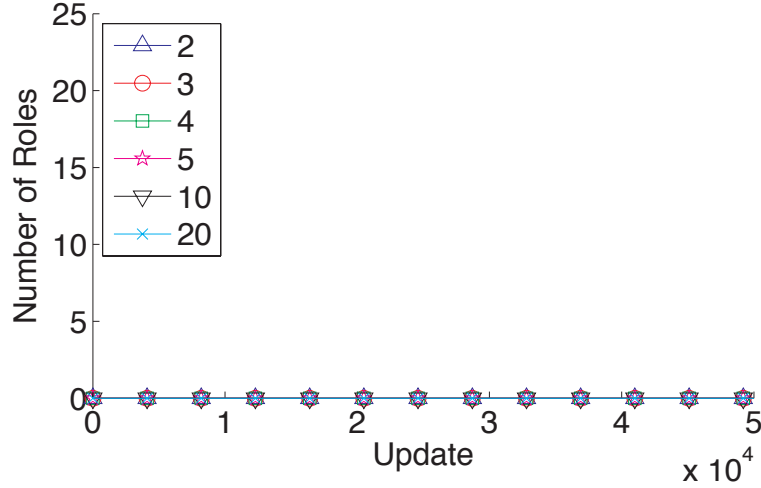
(Figure 3.7(b)).

However, for 5 or more roles, even the best groups were unable to perform division of labor. We found no statistically significant differences among these treatments at the $p < 0.05$ level (Kruskal-Wallis multiple comparison). Thus, it is clear that within this environment the group-level selection pressure is necessary to encourage diversity of roles. These two experiments have shown that this configuration of individual and group-level pressures are sufficient to evolve a simple form of division of labor, where all roles are rewarded equally. We now turn to more complex scenarios, where roles are rewarded unequally.

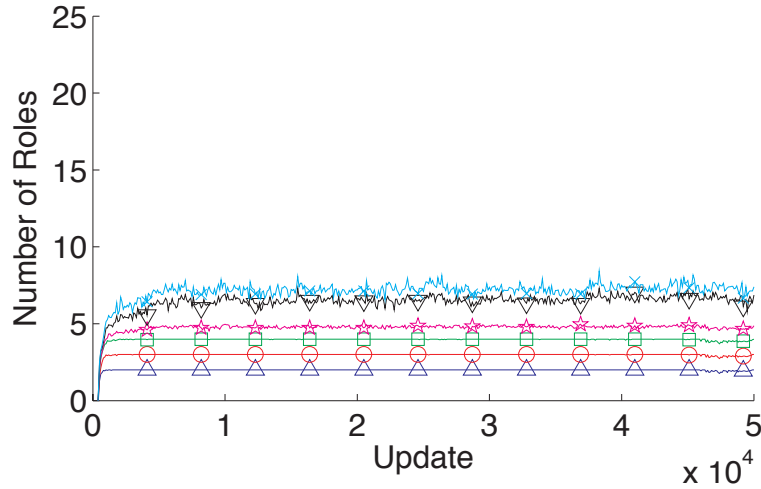
We conducted treatments where we set the desired number of role-ids to be five (R_5) and varied the distribution of the benefits among roles. Specifically, we conducted experiments: (1) where the benefits increased (multiplicative merit rewards of: 2, 3, 4, 5, and 6); (2) where one role was rewarded significantly greater than the others (rewards of: 2, 2, 8, 2, 2); and (3), where the majority of task bonus values were penalties (rewards of: 1, 0.75, 0.5, 0.75, 1). We again used the group fitness function described by Equation 3.8. In all cases, the most fit evolved group performed all five roles with the average group performing three or more roles. These results confirm that multilevel selection is able to produce groups in which individuals coordinate to perform different roles, even when the group and individual selection pressures are in conflict.

Can groups evolve to perform division of labor with complex roles?

For the last set of experiments, we studied whether this multilevel selection technique was sufficient to evolve division of labor when the complexity of the corresponding tasks varied. In our previous Avida experiments, an organism assumed a role by setting its role-id to a specific value. For these next experiments, we required the organism to perform a bit-wise logic operation on 32-bit integers. We used five mutually-exclusive logic operations: **not**, **nand**, **and**, **orn**, **or**. Specifically, an organism was rewarded once for performing one task



(a) Grand mean number of roles performed



(b) Mean of max number of roles performed

Figure 3.7: The results of using just individual selection to evolve groups that comprise varying numbers of roles, where each line represents a different treatment with a different desired number of roles ranging from 2 to 20. In these control treatments, the group-level component of multilevel selection was removed. Within this figure, (a) depicts the grand mean and (b) depicts the mean of the maximum number of unique roles over all 30 trials. Without the group-level component of the multilevel selection pressure, the groups were unable to evolve to perform multiple roles on average and in the best case evolve to perform 7 roles.

during its lifetime. If an organism performed multiple tasks, it was rewarded only for the first task it performed. In this way, we required organisms to be specialists. The challenge presented by this environment is that organisms must coordinate their roles in order to

specialize on different tasks. For the group competition, we evaluated the diversity of tasks performed by the current organisms in the group.

To examine whether division of labor occurred when the tasks were complex and their rewards were unequal, we rewarded the tasks in order of increasing complexity, which is calculated by the number of **nand** operations necessary to perform the logic task. Tasks **not** and **nand** (the simplest tasks) were assigned a bonus multiple of 2, tasks **and**, **orn** were assigned a bonus multiple of 3, and **or** (the most complex task) was assigned a bonus multiple of 4. The group’s fitness was simply the number of unique tasks performed by organisms within the group. Thus, similar to the other experiments, organisms have a pressure to perform the most highly-rewarded task, whereas the group has a pressure to perform all of the tasks. This treatment increases the difficulty of the division of labor problem because organisms must both evolve to perform logic operations and coordinate roles with different benefits. We refer to this treatment as the *Logic-5 Environment*.

Figure 3.8 depicts the results of the Logic-5 experiment. Here, Figure 3.8(a) shows the grand mean of the number of unique roles, while Figure 3.8(b) displays the mean of maximum number of unique roles. In addition to the experimental treatment, the results of two control treatments are shown. For the experimental treatment (**Logic-5**), both the individual and group-level selection components of the pressure were present. On average, groups performed over 4 tasks; the best performing group performed all 5 tasks. These results indicate that the multilevel selection pressure is sufficient to produce division of labor, where, within a given group, organisms evolve to coordinate the performance of complex disparate roles.

The control treatments depicted within Figure 3.8 demonstrate the effects of removing various selection pressures. For the first control treatment, only the individual-level pressure is present (**Ind, no group**). The average group performed less than one task; the best groups performed an average of under three tasks. The second control treatment (**No group; no ind**) has no selection pressures – we removed both the group and the individual-level selection.

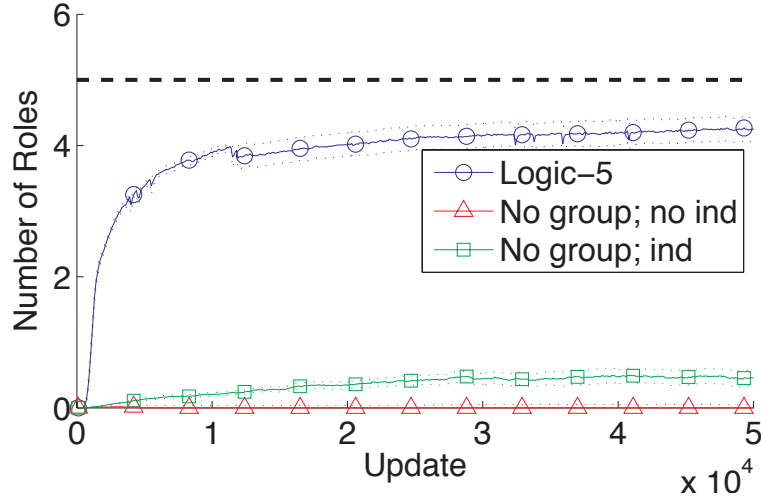
The average group performed zero tasks; the best groups performed an average of 1 task. (We did not perform a control with just group selection pressure, since this simplifies the division of labor task and was done in Section 3.3.2.) These controls demonstrate that the group-level component is, as expected, a critical force in maintaining division of labor.

3.3.3 Analyses

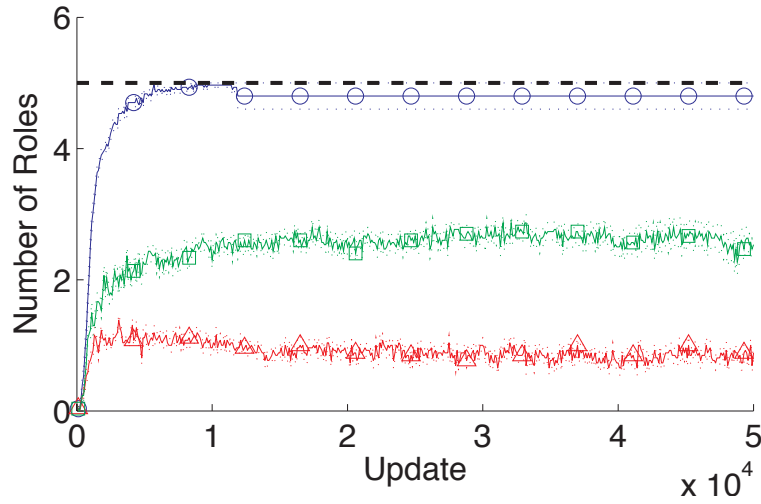
We have demonstrated that multilevel selection can produce groups where the organisms specialize to perform different roles, even if these roles are rewarded unequally and vary in complexity. Next, we examine how division of labor arose within these groups. The first question addressed by our analysis is whether the groups evolved a heritable strategy for division of labor. The second question is if the groups evolve to perform division of labor, then how do they maintain the coexistence of organisms that have different associated fitnesses? In other words, what prevents the organism(s) with the greatest fitness from sweeping the group? We address both these questions within the Logic-5 environment (described in Section 3.3.2). Finally, we conduct a detailed analysis of one group that performed all five tasks.

Do groups evolve a heritable strategy for performing division of labor?

A central concern is whether or not the performance of the groups indicates that they evolved a heritable strategy for division of labor, or whether what we perceive to be a strategy is simply an artifact of the experimental design and the performance monitoring technique used. To begin our investigation, we identified two alternative hypotheses that are summarized in Table 3.2. These hypotheses explain why the observed behavior could have been something other than an evolved division of labor strategy.



(a) Grand mean number of roles performed



(b) Mean of max number of roles performed

Figure 3.8: Control and experimental results of using multilevel selection pressures to evolve groups of organisms that perform five different logic tasks with varying rewards. The dashed line represents the best possible performance. Here, (a) depicts the grand mean number of roles performed and (b) depicts the mean of max roles performed across 30 trials. The two control treatments (**No group; no ind** and **No group; ind**) each evolve to perform fewer than 3 roles; whereas, the experimental treatment with both components of the multilevel selection pressure evolves to perform 4 roles on average and 5 at best. These results are evidence that multilevel selection pressure is sufficient to produce groups that perform five logic tasks with varying rewards.

Alternative Hypothesis 1: Division of labor can be explained by mutations. Our first hypothesis (Table 3.2) is that division of labor is a result of mutation-selection balance,

Hypothesis	Predictions tested
1. Apparent division of labor is actually a result of mutations occurring within groups re-creating the observed diversity.	Removing mutations will result in a decrease of division of labor in an evolved group.
2. Performance of diverse tasks is probabilistic and evidence of division of labor is an artifact of group-level selection continuously selecting the most diverse groups.	Removing group selection pressures will result in an immediate loss of division of labor in an evolved group.

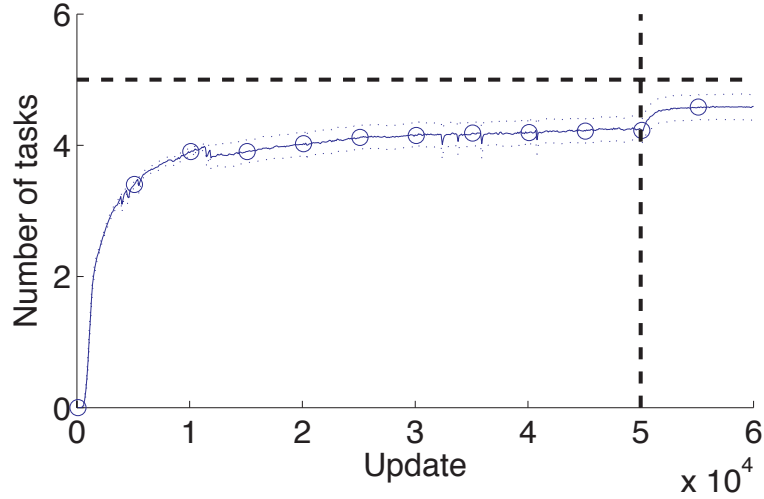
Table 3.2: Alternative hypotheses and predictions tested in this study.

where organisms are attempting to perform the most highly-rewarded task and perform other less-rewarded tasks as a result deleterious mutations. This dynamic is possible in the Logic-5 environment because the tasks are building blocks for one another [72]: complex tasks frequently evolve using pieces of the simpler tasks. Thus, a deleterious mutation to a complex task may result in an offspring organism performing a simpler, less-rewarded task. If this hypothesis is correct, then when mutations are turned off, the organisms performing the most highly-rewarded tasks should sweep the group. Because organisms performing the less-rewarded tasks will not exist, there will be an overall decrease in division of labor.

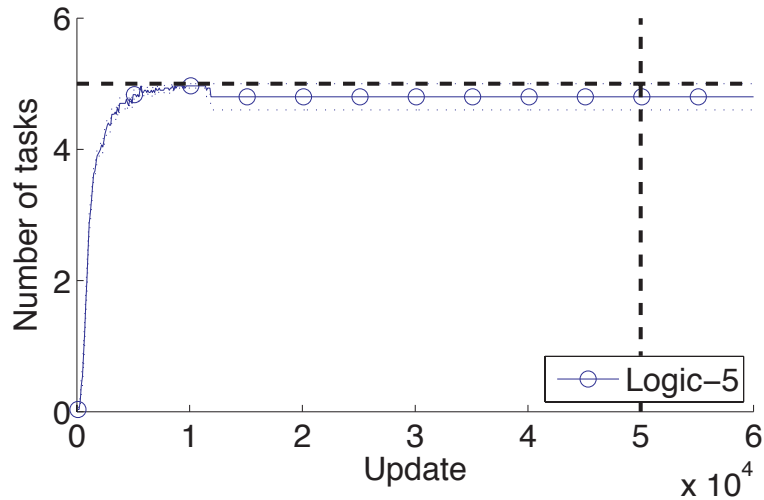
To test this hypothesis, we extended the original experiment with a 10,000 update *ecological phase*. In Avida, an ecological phase is a period following the completion of an experiment during which no further random mutations can occur. Ecological phases eliminate spurious genotypes that occur as the result of deleterious mutations that are not able to fix in the population. In this case, we use the ecological phase to ensure that the strategy adopted by a group is able to perform division of labor without random mutations. During the ecological phase, we performed group competitions and recorded the diversity of tasks performed by the group using the same methods as the original experiment.

Figure 3.9 depicts the grand mean and maximum performance of the groups over time, including the ecological period; all data are averaged across 30 trials. Mean performance *improves* during the ecological period, indicating that groups are more consistently able to

achieve division of labor in the absence of mutations. Moreover, the maximum performance of the groups remains constant from 50,000 updates to 60,000 updates, indicating that mutations are not required for division of labor to be maintained.



(a) Grand mean number of roles performed



(b) Mean of max number of roles performed

Figure 3.9: The average mean (a) and average maximum (b) performance of the groups through the normal 50,000 update period and then a 10,000 update ecological phase, where mutations are turned off. The horizontal dashed line represents the maximum number of tasks possible and the vertical dashed line represents the start of the ecological period. The dotted lines represent standard error. The mean group performance increases and the maximum group performance does not decrease from 50,000-60,000 updates, indicating that mutations are not a key part of the group strategy.

Additionally, we further analyzed the performance of the 29 *dominant groups*. To identify the dominant groups, we selected one of the best-performing groups from each of the 29 trials that finished. In run 2, all the groups died prior to the completion of the run, thus all of its values are zero and we do not include it in our reports of the analysis results. Each of these 29 groups performed all 5 tasks during the last 100 update period of the original experiment. To confirm that mutations were not affecting the strategy of the dominant groups, we ran each dominant group in isolation 30 times for an ecological period of 100 updates, with different random number seeds, and tracked their average performance (a total of 870 tests). Each group was tested multiple times to account for potentially stochastic strategies. The groups, on average, maintained 4.44 tasks after the ecological period ($se \pm 0.08$). This serves as further confirmation that division of labor cannot be explained by mutations alone.

Intriguingly, while examining the data to better understand how mutations were used as part of a division of labor strategy, we uncovered an unusual result: At the end of the ecological period, on average, 10% of offspring differed from their parents. Given that the mutation rate was set to 0 during the ecological phase, some organisms evolved *unstable genotypes*, where their replication cycle produced offspring with a different genome. Furthermore, during our analysis, we discovered that several of the dominant groups had evolved unstable genotype strains that could potentially help them reinforce diversity to achieve division of labor. For example, in one interesting case a parent with an unstable genotype sliced its genome in half and inverted the two parts to produce a progeny that performed a different task. Of course, the organism's inverted offspring is unlikely to be able to replicate, but it would aid in the overall task-diversity of the group. This is akin to sterile workers within eusocial colonies that assist in the completion of group tasks, but do not create offspring. Thus, although the groups do not rely on external mutations to perform division of labor, unstable genotypes may be part of evolved and inherited strategies.

Alternative Hypothesis 2: Division of labor can be explained by group-level selection. Our second alternative hypothesis (Table 3.2) is that the groups did not evolve heritable strategies for division of labor, but rather diversity is unstable and is continually being propped up by group-level selection. For example, a group that uses an unstable strategy might have five different genotypes, each of which performs a different task, where individual selection would typically cause one of the genotypes to eliminate the other four. Considering that all organisms start each competition period with equivalent merit (and thus execute virtual CPU instructions at the same rate), this most-fit genotype may fail to sweep before the groups compete again. Thus, the group would maintain diversity, but this diversity would occur as a failure to sweep rather than intentional maintenance of organisms that perform lesser rewarded tasks. Groups that happen to maintain diversity in this manner would then get selected for giving the incorrect impression that diversity is stably maintained.

If this hypothesis is correct, then, as a result of stochastic performance differences, a given group would not consistently perform the same number of tasks. To assess whether group performance was consistent, we ran each dominant group (selected in the same manner as in Alternative Hypothesis 1) in isolation 30 times for 100 updates, with different random number seeds, and tracked their average performance (a total of 870 tests). On average, the groups performed 4.44 tasks ($se \pm 0.07$), indicating that the performance of the groups is consistent and is not obscured by group-level selection pressures. Thus, the performance of division of labor cannot be accounted for by group-level mutation selection balance.

Because both hypotheses regarding potential mechanisms for the groups to exhibit several tasks and yet not perform division of labor have been refuted, there are indications that the groups have instead evolved a heritable strategy for coordinating and performing the various logic tasks.

How do groups perform division of labor?

Given that the groups appear to evolve a heritable strategy for performing division of labor, we next examine the components of their strategy. Specifically, how do the groups coordinate their roles and manage to maintain organisms that perform the less-rewarded roles? Two potential components of their strategy are: (1) explicit coordination resulting from the integration of instructions we provided and (2) the maintenance of genetic diversity within groups through a carefully regulated replication process. Here we explore whether these two mechanisms are used.

Do the groups use instructions to coordinate roles? For our division of labor experiments, we provided instructions that enabled explicit coordination using messaging and location-awareness. A group could use these instructions to coordinate roles. For example, organisms could select a task based on their location or could send messages alerting other organisms of their selected task. To better understand how the behavior of a group is influenced by the instructions within its constituent genomes, we conducted *knockout experiments*, where we replaced one or more instructions in all of the genomes in a group with a neutral instruction (called **nop-X**). We then subjected the group to an ecological period and monitored its behavior. If the group’s ability to perform division of labor was impaired following a knockout, then we can conclude that the knocked-out instruction contributed to its behavior.

To explore whether groups are using explicit coordination as part of their strategy, we ran a series of knockout experiments on the dominant groups. We performed three different sets of knockout experiments: (1) Knockouts of location sensing capabilities (i.e., the **get-cell-xy** and **collect-cell-data** instructions). (2) Knockouts of messaging capabilities (i.e., the **send-msg** and **retrieve-msg** instructions). (3) Combined knockouts of both types of coordination capabilities.

Group division of labor performance was impacted by the knockout experiments for communication and/or location awareness. Without location sensing capabilities, the groups performed an average of 3.12 tasks ($se \pm 0.07$); without messaging capabilities, the groups performed an average of 3.10 tasks ($se \pm 0.08$); without coordination capabilities of any kind, the organisms performed an average of 3.14 tasks ($se \pm 0.10$). All of these are appreciably lower than the 4.44 tasks performed on average after the ecological period. However, the groups are still performing the majority of the tasks. Overall, based on the behavior of the dominant groups in the knockout experiments, we conclude that the coordination instructions were used in a limited capacity to maintain division of labor. We turn our attention to exploring other mechanisms the evolved strategies could have used to preserve their ability to perform a diversity of tasks without the coordination instructions.

Do the groups use genetic diversity to coordinate roles? Because group replication events copy all organisms into the new group, it is possible that the groups could contain multiple co-existing species, where each species is responsible for performing a different role. However, if this is the case, how are organisms that perform the lesser rewarded task(s) maintained within the group? In contrast, if the organisms within a group were genetically identical or *clonal*, then performing a low-rewarded role would not affect the ability of the organism’s genetic material to survive, but they would need an additional coordination mechanism. Here we examine whether genetic diversity was present within the groups, and if so, how it was maintained.

As a basis for understanding the genetic diversity present within the trials, we compare the genetic diversity of the dominant groups (i.e., the best performing groups from each of the trials) to the genetic diversity of the dominant groups from the control runs (described in Section 3.3.2), where the groups had varying individual-level pressures, but no group-level pressure. To directly measure the genetic similarity within a group, we calculated the

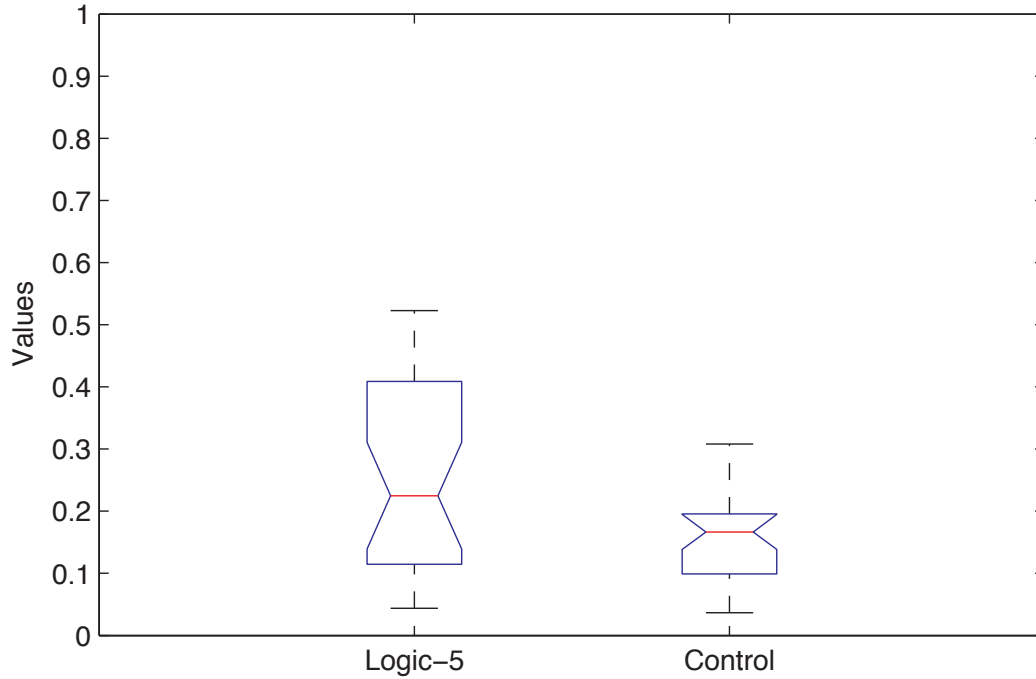


Figure 3.10: The percent of genetic diversity in the dominant Logic-5 groups and control runs. Notably, the Logic-5 groups have more genetic diversity even though they also must maintain this diversity while performing tasks with different associated benefits.

average *Levenshtein distance*, which is the number of insertions, deletions, and substitutions necessary to convert one genome into another [74], normalized by genome length, for all pairs of organisms in a group. Figure 3.10 depicts the results for the dominant groups from the Logic-5 and control runs. As can be seen here, in the Logic-5 environment, each pair of organisms within a group differed by, on average, slightly more than 25%. For the control groups, each pair of organisms within a group differed by, on average, slightly more 15%. Notably, the genetic diversity in the Logic-5 groups was statistically significantly greater than the genetic diversity in the control groups (Mann-Whitney U Test $p < 0.05$). This result indicates that genetic diversity is present and thus may be an aspect of the group-level strategy for preserving division of labor.

These data enable us to conclude that genetic diversity is present within the groups. What we do not know is whether this diversity is intentionally maintained and how it may affect

division of labor. One possible mechanism for organisms to maintain genetic diversity within the group is to modify their rate of replication so that all the lineages are maintained. There are two components to an organism's replication rate: (1) its merit, which is determined by the first task it performs, and (2) its gestation period. While an organism's merit is dependent upon the first task it performs, its gestation period is dependent upon the efficiency of the organism's replication loop, the number of instructions in its genome, and its execution flow. Given different organisms may have different gestation periods, it is possible for the organisms within a group to evolve such that more highly rewarded tasks also have a longer gestation period: although they may run faster, they also perform more instructions prior to replicating. To stably maintain genetic diversity, it is possible that organisms within groups have adjusted their gestation cycles to produce offspring at the same rate. To study the role of genetic diversity within these experiments we focus on the behavior of one successful group in the subsequent case study.

Case Study

Here we perform a detailed analysis of one group. We selected group 29 for additional analysis because it is the highest scoring group after the ecological period, it contains unstable genotypes, and its performance is affected when communication and location instructions are knocked out.

After one ecological period (with a seed selected at random), this group performed all 5 tasks. It included 15 different genotypes, where 10 genotypes were present in the original group and 5 were produced as a result of unstable genotypes. Figure 3.11 depicts the genotypes present at the end of the ecological period, where the shaded instructions represent the coordination instructions. We clustered these genotypes into two ecotype families based on genetic similarity, which is visible by the different banding patterns created by the coordination instructions. Although the members of each ecotype were quite similar, representative

members from each of the two families differed significantly enough to easily identify them by visual inspection.

To identify the specific roles performed by each ecotype, we analyzed each genotype in isolation. Specifically, we filled a group with each genotype and subjected it to 30 ecological periods of 100 updates. This process revealed that the members of one ecotype family were capable of performing only the highest rewarded task (i.e., **orn**). The members of the other ecotype family were capable of performing all four lower rewarded tasks (i.e., **not**, **nand**, **and**, **orn**, or). Moreover, this second ecotype family comprised the unstable genotypes. By comparing organisms to their mutated offspring we were able to determine the nature of these implicit mutations. In this case, the unstable genotypes removed an instruction from the start of the genome. Within Figure 3.11 the second cluster of genotypes are members of the unstable genotype ecotype. We have organized the genotypes to make the pattern of implicit mutations more clear. As the genotypes within this family are read from left to right, instructions are missing from the start of the genome resulting in the appearance of the banded coordination instructions ‘moving up’ the genome. These missing instructions do not appreciably change the behavior of the organisms.

Overall, the two ecotypes present in the group fill mutualistic ecological niches. Each ecotype performed one or more tasks that the other did not perform. Together they were able to perform all 5 tasks. One interesting aspect of these individual runs is that they supported our hypothesis that the groups were using their replication cycle to facilitate diversity. Almost all of the different genotypes had between 2 and 3 generations within the 100 update ecological period. Thus, despite the differently rewarded tasks, no genotype reproduced so rapidly that it swept the population, destroying members of the other ecotype.

Next, to understand the behavior of the group as a whole, we analyzed one ecological period to identify which genotypes were present and which tasks they performed. Figure 3.12 depicts the genotypes and the task they performed during the ecological period. The shading

Figure 3.11: The 15 genotypes present in group 29 after an ecological period of 100 updates. Each column is one genotype. Each column element is an instruction. The coordination instructions are colored. There are two different ecotypes present within this group. They are distinguishable by the different patterns of banding.

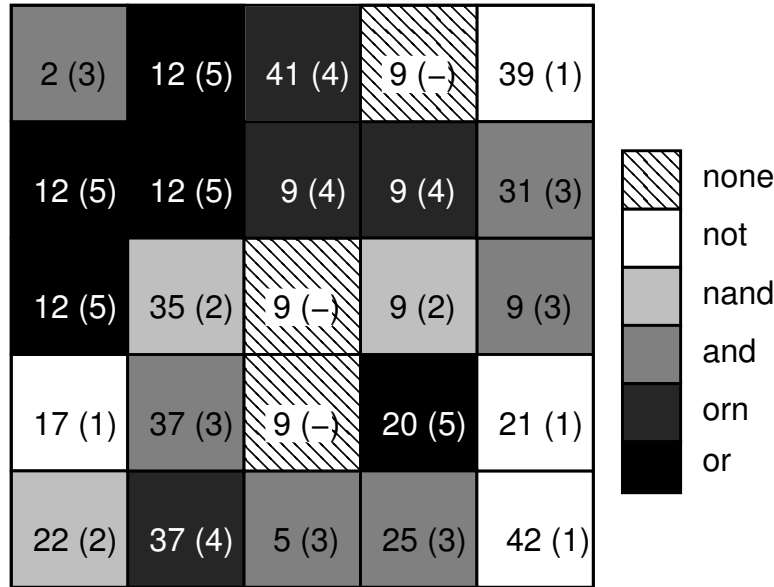


Figure 3.12: A visual depiction of the genotype and phenotypes of the organisms that comprise group 29. The shading represents the task performed (phenotype). The first number represents the genotype and the second number represents also denotes the task performed (phenotype). The black squares are the organisms in ecotype family 2. The remainder of the squares are in ecotype family 1.

is used to indicate the task performed and the label specifies the genotype and the task performed in parentheses. A dash indicates that the organism did not perform a task. During this period, all 5 tasks were performed. Specifically, **not** was performed 4 times, **nand** was performed 3 times, **and** was performed 6 times, **orn** was performed 4 times, and **or** was performed 5 times. The first ecotype comprises genomes 12 and 20. The second comprises the rest of the genotypes. Notably, several genotypes from the second family exhibit phenotypic plasticity. For example, genotype 9 performs no tasks, task 3 (**and**), and task 4 (**orn**). Genotype 37 performs task 3 (**and**) and task 4 (**orn**). Based on this analysis, group 29 uses genetic diversity in concert with careful replication rate timing to ensure all five tasks are performed by organisms from two different ecotypes.

3.3.4 Discussion

In this study, we demonstrated that multilevel selection pressures can be used to evolve cooperative, genetically heterogeneous groups, where individual roles have varying fitness benefits. We examined the strategies of groups that successfully evolved to perform division of labor. The key components of their strategy were genetic diversity sustained through ecological niching and the use of coordination instructions.

This work ties into our larger objective of using digital evolution as a platform for researching the evolution of division of labor. Here, we examined how organisms coordinated their roles, and whether division of labor could evolve within heterogeneous groups. Next, we examine the pressures that result in the evolution of division of labor within genetically homogeneous (clonal) groups.

3.4 Division of Labor: Clonal Groups

Thus far, we have focused on genetically-heterogeneous groups. One reason for this focus is that our early studies were motivated by evolutionary psychology and human groups, such as business organizations, that frequently comprise members that are not genetically related. However, human behavior, especially the evolution of human behavior, is exceedingly complex and challenging to study. Thus, for the subsequent studies we focus on genetically clonal groups. Clonal groups, such as some slime molds [8, 16, 98] and some colonies of aphids [92], can be found within nature and also can be useful as a model for understanding the behavior of other eusocial organisms. The main advantage of clonal groups is that they focus on questions central to division of labor and complications, such as the coexistence of species and altruism. However, one complicated aspect of division of labor studies with clonal groups is that for them to exhibit division of labor they must use the same genotype to produce several different phenotypes.

3.4.1 Methods

For these experiments, we use the default group topology, where the groups could contain up to 25 clonal organisms. Specifically, to ensure genetic homogeneity, when an organism replicates within a group its genome is not mutated. These groups competed every 100 updates in tournament selection, where fit groups reproduce using germ-line replication. For these experiments, we continue to use mutually exclusive tasks and make use of the 25-role environment and logic environment, which were previously described. We ran 30 trials for each experiment.

3.4.2 Results

Our preliminary studies address two questions related to the evolution of division of labor [44]: First, if we require the organisms to be *specialists* that perform only one task each, then can clonal groups 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? We examine what mechanisms the groups evolved to perform division of labor in both environments.

Can clonal groups perform division of labor?

To assess whether clonal groups are able to evolve to perform division of labor if specialization is required, we repeat versions of the role and logic experiments that we performed for genetically-heterogeneous groups [43, 45] described in Section 3.3. To enable genetic homogeneity within groups, we used germlines (described in Section 2.1.5) to construct each of the groups. Additionally, because the groups are clonal, we subject them only to group-level pressure, rather than multilevel selection.

First, we assess if organisms in groups are able to evolve an extreme form of division of labor, where each organism is required to perform a single distinct role. These tasks and the group fitness function are the same as those used for the genetically heterogeneous groups described in Section 3.3. Specifically, the target role-ids are 1 through 25 and groups are rewarded for performing a diversity of roles. 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 group competition period, creating an explicit pressure for division of labor. Specifically, the group fitness function we used here is described by equation 3.8.

Figure 3.13 depicts the maximum (blue line with circles) and mean (red line with triangles) number of tasks performed by the groups averaged across 30 runs. On average, the mean group performance was approximately 14.86 (standard error ± 0.67) tasks. However, the organisms within the best groups specialized to perform 23.63 (se ± 0.33) different tasks. These results indicate that groups of organisms are able to evolve effective strategies for both specializing to perform roles and also coordinating the distribution of roles among members of the group.

Our second set of experiments assess whether organisms in groups are able to evolve division of labor if the roles require more complex computation. For these experiments, we use a variation of the standard Avida logic environment. Each role is a task that requires 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 [87]: **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 tasks are more challenging to evolve, but also require less coordination, since there are nine tasks and 25 organisms in a group. The group fitness function we used

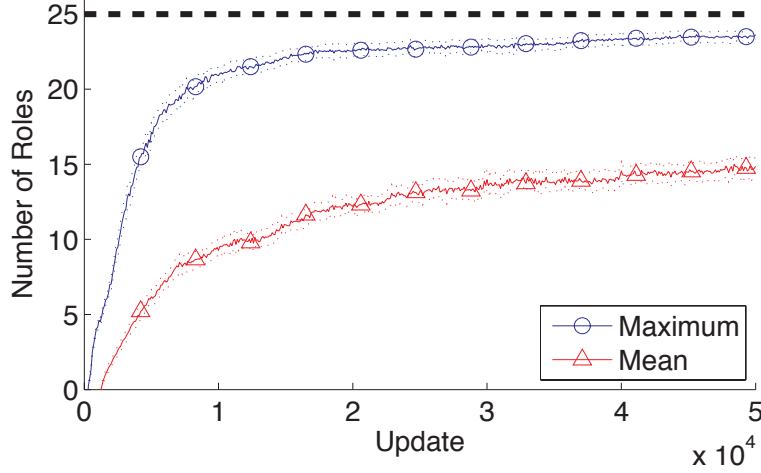


Figure 3.13: The mean and maximum number of roles performed by groups within the 25-role environment across 30 trials. Notably, some groups approach a perfect score (denoted by a dashed line) of 25.

here was simply:

$$F = 1 + r \quad (3.9)$$

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

Figure 3.14 depicts the maximum and mean number of tasks performed by the groups averaged across 30 runs. On average, the best group performed 6.17 ($se \pm 0.16$) tasks and the average group performed 4.27 ($se \pm 0.10$) tasks. These values indicate that the problem is challenging, but that the groups are still able to evolve to perform division of labor with complex tasks. We note that when the group competition pressure is removed, no tasks were performed in any group, indicating that genetic drift alone is not sufficient to produce division of labor (data not shown).

What mechanisms do clonal groups use to perform division of labor?

Given that clonal groups of organisms were able to evolve to perform division of labor, we next investigated the mechanisms they use to assign roles and whether these mechanisms

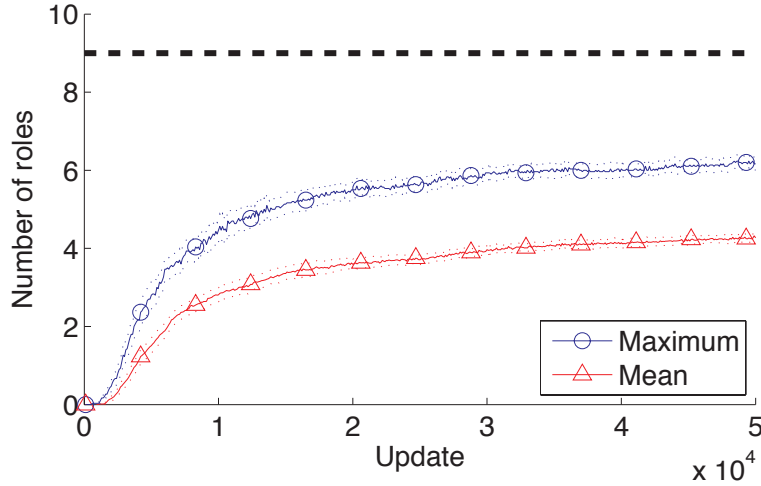


Figure 3.14: The mean and maximum number of roles performed by groups in the logic environment, where there are nine tasks of increasing complexity. The best possible performance is denoted with the dotted line. Groups on average perform four tasks and at best perform 6 tasks.

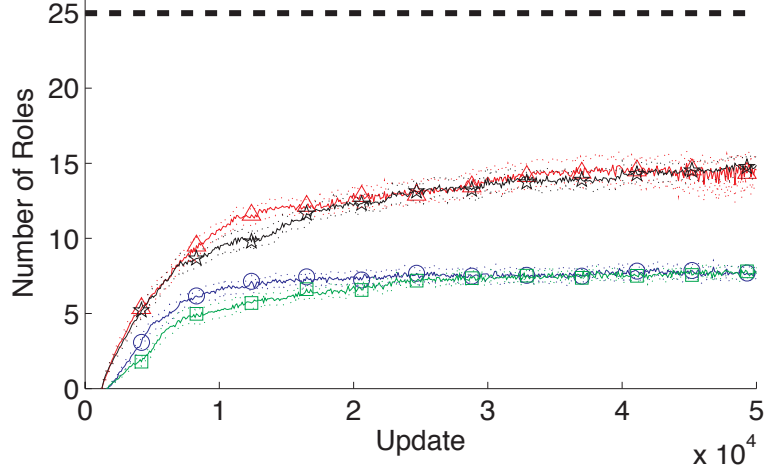
change based on environmental context. For the original experiments, we provided the organisms with two different types of mechanisms that we expected them to use to differentiate: message-based communication and location information. Specifically, we provided instructions that, if evolved into a genome 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 group (`get-cell-xy`) and to access cell data, a random 32-bit integer associated with their location (`collect-cell-data`). These mechanisms are similar to those used by groups within nature. For example, cells within multicellular organisms use chemical gradients, which are an abstract representation of location, to determine their fate. Insects within eusocial colonies exchange messages to notify others of which tasks need to be performed [53].

To better understand how the mechanisms we provided to the organisms enabled them to evolve division of labor in different environments, we performed additional experimental treatments that removed one or more coordination capabilities. For each environment, we

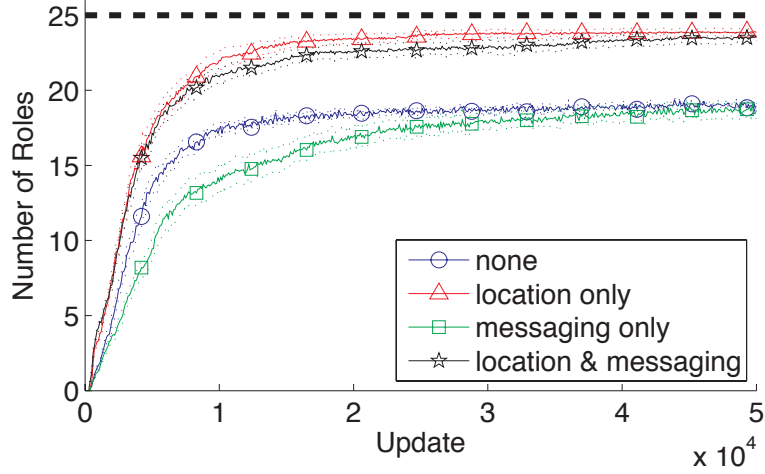
performed three additional treatments: (1) The organisms had location information, but no messaging capabilities (**location only**). (2) The organisms had messaging capabilities, but no location information (**messaging only**). (3) The organisms were deprived of messaging capabilities and location 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.

First, we examined how the organisms performed division of labor in the 25-role environment. Figure 3.15 depicts the mean (a) and maximum (b) performance of the groups 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 groups are ignoring messaging and instead are using their x and y coordinates to perform division of labor.

Next, we examined how the groups 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). These ambiguous data 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. This stochastic strategy is clever and also biologically realistic – many cells within multicellular organisms take advantage of random processes to select their cell fate. However, for the purposes of this study, we were more interested in which mechanism they would use if there was less randomness within their environment. Thus, in an attempt to compel the organisms to evolve a non-stochastic strategy, we deprived them of two of the obvious sources of random



(a) Grand mean number of roles performed



(b) Mean of max number of roles performed

Figure 3.15: The mean (a) and maximum (b) number of roles performed in the 25-role environment when different combinations of coordination instructions are removed. For this environment, group performance depends upon the availability of location information.

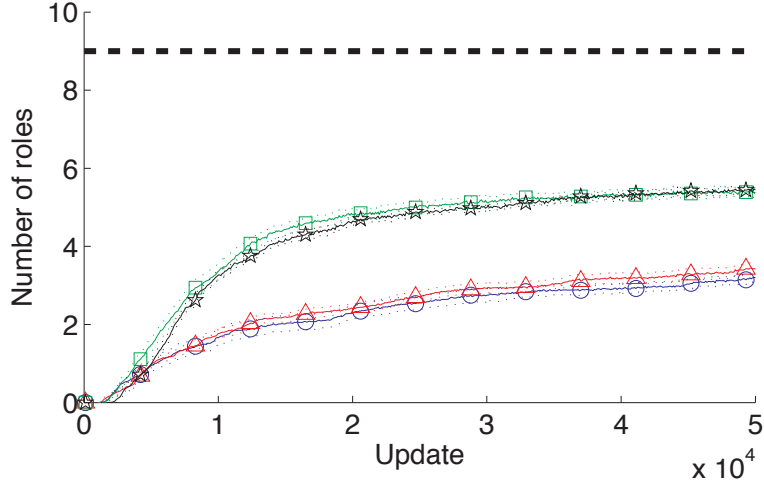
information (i.e., random inputs and random cell data) and repeated all four treatments.

Figure 3.16 depicts the mean (a) and maximum (b) performance of the groups 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 3.14) in that random information has been removed. Interestingly, the **location & messaging** treatment that does not include random information significantly outperforms the original

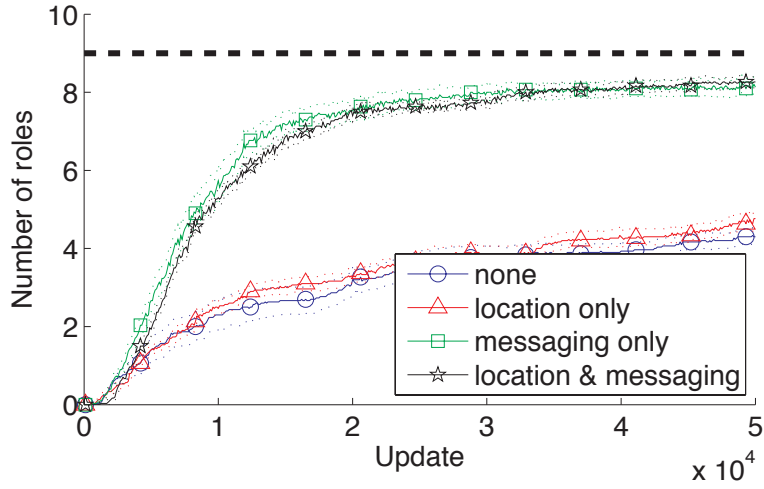
treatment that does. The mean and maximum group performance are 5.44 (se \pm 0.10) and 8.27 (se \pm 0.14) tasks for the **location & messaging** treatment, as compared to 4.27 (se \pm 0.10) and 6.17 (se \pm 0.16) tasks for the original treatment. This disparity indicates that random information presented a division of labor mechanism that was easier to evolve, but 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. Next, we analyze potential reasons for this difference.

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

One open question is why the mechanism used by the groups differed between environments. We identified two possible hypotheses for why organisms evolved to use messaging capabilities in the logic environment and location information in the 25-role environment.



(a) Grand mean number of roles performed



(b) Mean of max number of roles performed

Figure 3.16: The grand mean (a) and mean of maximum (b) number of roles performed in the logic-9 environment when different combinations of instructions are removed. For this environment, group performance depends upon the availability of messaging capabilities.

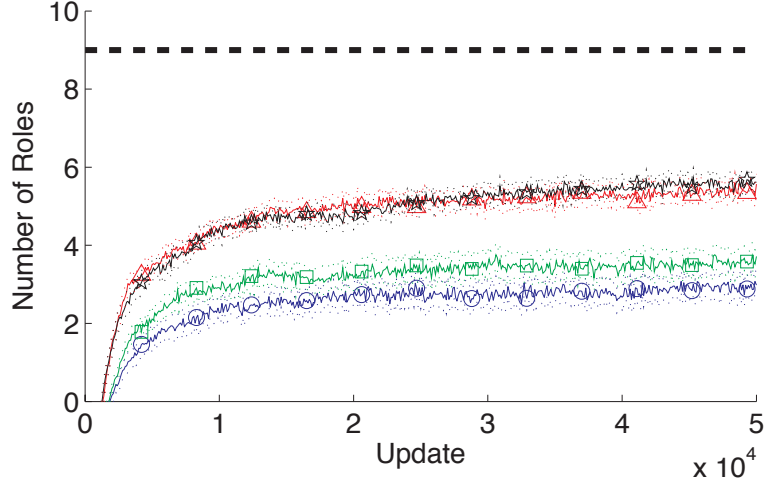
The first hypothesis is that the type of task influences which coordination mechanism is preferred. The second hypothesis is that the number of tasks influences which coordination mechanism is preferred. Because the two experiments differ both in terms of number and type of task, we needed to perform 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

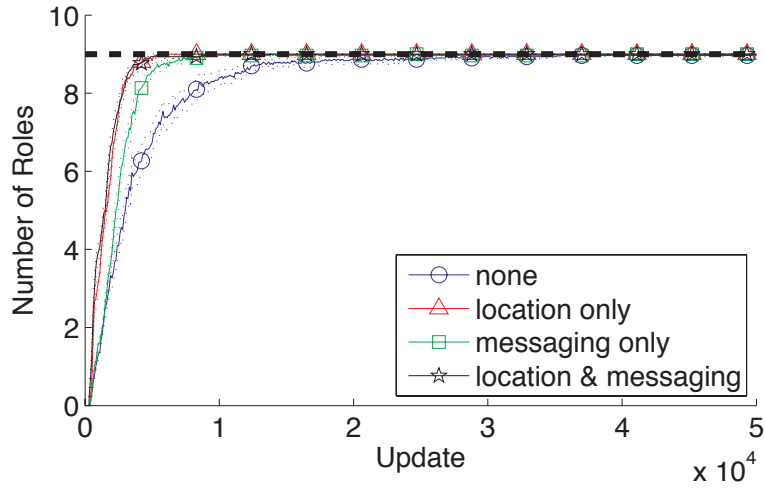
nine different roles. Organisms could fill these roles by setting their role-ids to the numbers 1-9. This environment used the same type of tasks as the 25-role environment, but had the same number of tasks as the logic environment. If the groups evolved to use messaging, it would falsify our hypothesis that the type of task affected the mechanism of division of labor. Conversely, if the groups evolved to use location information, this outcome would falsify our hypothesis that the number of tasks is the key factor in selecting a mechanism for division of labor. Figure 3.17 depicts the mean (a) and maximum (b) performance of the groups 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.

3.4.3 Discussion

In these preliminary studies, we demonstrated that groups 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. These studies demonstrate that Avida is able to evolve groups that exhibit division of labor.



(a) Grand mean number of roles performed



(b) Mean of max number of roles performed

Figure 3.17: The mean (a) and mean of maximum (b) number of roles performed in the 9-role environment when different combinations of instructions are removed. For this environment, group performance depends upon the availability of location information indicating that the type of tasks is more important than the number of tasks.

Chapter 4

Forces Driving the Evolution of Division of Labor

In this chapter, we leverage the results of our previous studies to explore the conditions under which division of labor evolves when it is not directly selected for. We address two central questions: (1) Do organisms specialize in different tasks at different points in their life (temporal polyethism) due to the risk factors associated with those tasks? (2) Do increased task-switching costs result in an increase in division of labor?

4.1 Temporal Polyethism

Many eusocial insects exhibit temporal polyethism, where a worker's age is correlated with the type of task it performs [31, 53, 102, 107, 119, 120, 121]. For example, within honeybee colonies, worker bees progress sequentially through four castes: cell cleaning caste, broodnest caste, food storage caste, and forager caste [106]. Within ant colonies, a shift is similarly performed from activities within the nest, such as brood care, to foraging activities outside the nest [53]. Researchers are still actively exploring the causes of this division of labor

pattern. Two theories have been proposed. The first theory, is that an individual’s age is causally linked to the task that it performs [53, 102, 121]. This causal relationship is thought to result from an internal development program that was selected for as a result of the relative risks associated with tasks and aging. Specifically, more risky tasks are performed when the organism is older and closer to death. For example, foraging, which is the most risky task and is responsible for the loss of 1% to 10% of the colony population per day [53], is performed when the organism is likely to die of natural causes and thus is more expendable. In this way, the colony optimizes the use of its organisms [119]. The second theory, called *foraging for work*, is that the temporal polyethism pattern, while correlated with age, results from the organization of colony nests [31, 53, 107, 120]. As organisms are born, they perform tasks closest to them and proceed to perform tasks further and further from the center of the nest [107, 120]. The available evidence regarding the behavior of eusocial insects makes it difficult to decide between the two theories [31, 53, 102, 121]. Additionally, while studies can observe the mechanisms used by modern colonies, they cannot explore the evolutionary conditions that may give rise to the temporal polyethism pattern. In this study, we use Avida to test whether the causes proposed by the first theory, age polyethism, are sufficient for evolving colonies that exhibit temporal polyethism.

4.1.1 Methods

To use Avida to study the evolution of temporal polyethism, we created a world consisting of competing groups that represent colonies. These groups comprise a set of clonal organisms that are rewarded for completing a group task. Specifically, the group must perform a set of tasks a number of times each in order to perform germ-line replication. For example, in our initial experiments, a group had to perform task **not** 250 times and task **nand** 250 times. A natural analog is that colonies of eusocial insects must both forage for food and tend to the brood. In addition, because each colony starts with only one organism, organisms must

also replicate to produce other organisms that can assist them in the performance of tasks to achieve the overall group objective.

To address our central question regarding the evolution of temporal polyethism, we extended Avida to support task riskiness. Specifically, we added the capability for each task to be associated with a *lethality risk* that specifies the probability of the organism dying while completing the task. Non-risky (or safe) tasks have a lethality risk of 0. Our most risky tasks have a lethality risk of 25%.

Additionally, we configured Avida to make use of existing aging configuration options. First, to ensure an organism could perform each task multiple times within its life, we modified a configuration parameter to extend the age of an organism from approximately 600 cycles to approximately 1200 cycles. Second, once Avida organisms replicate, they are generally reset to their initial starting configuration (i.e., the organism’s registers are reset, its age is set to 0, and it resumes execution at the beginning of its genome). This feature is designed to emulate the behavior of bacteria that divide into two daughter cells when they replicate. However, since age and internal state could play a key role in these experiments, we modified the organisms so that they do not reset after replication, but rather just continue running.

At the outset of these experiments, we seed the groups with an *ancestor* organism that performs all the types of tasks necessary for completion of the group task. For example, for our initial experiment, an ancestor organism performs task **not** and task **nand** once. Each experiment comprises several different treatments that randomize the order in which the tasks appear in the ancestor organisms’ genomes, as well as the riskiness associated with the tasks. We will describe the specific configurations in greater detail when we introduce the experiments. For each experiment, we conducted 30 trials to account for the stochastic nature of evolution. Within each trial, we used the default group topology, where groups comprise clonal organisms and replicate as a result of group competition.

4.1.2 Results

Our studies address whether temporal polyethism results from the risks associated with aging and various tasks in a two-task and three-task environment.

Can temporal polyethism result from risks associated with tasks?

The primary topic of this study is whether the risks associated with aging and tasks are sufficient to evolve groups of organisms that exhibit temporal polyethism. For our initial study, we created a two-task environment in which colonies had to perform task **not** 250 times and task **nand** 250 times in order for the group to replicate. We created four risk treatments (described in Table 4.1) that vary the lethality risks associated with the tasks. Specifically, the treatments are: (1) task **not** is risky, (2) task **nand** is risky, (3) neither task is risky, and (4) both task **not** and task **nand** are risky.

Task	Risk Treatments			
	not risky	nand risky	Control: no risk	Control: both risky
not	25%	0%	0%	25%
nand	0%	25%	0%	25%

Table 4.1: The four risk treatments for a two-task environment. The rows describe the lethality risks associated with tasks **not** and **nand**. (E.g., A 25% risk means that while performing the task, the organism has a 25% chance of being killed prior to receiving credit for the task.) The columns describe a specific treatment.

Additionally, we created two possible ancestor organisms (depicted in Figure 4.1). Both ancestors complete each task and self-replicate. However, ancestor **not-nand** performs the **not** task first and ancestor **nand-not** performs the **nand** task first. By varying the ancestor organism, we are able to verify that any patterns of temporal polyethism result from the riskiness associated with the tasks, not the initial genomic structure of the organisms. For each ancestor, we performed all four risk treatments. If our hypothesis is correct and task riskiness is a sufficient pressure to result in temporal polyethism, then we should see that

organisms perform the less risky task first and the more risky task second, regardless of which task is more risky or the order in which the tasks are performed by the ancestor organism.

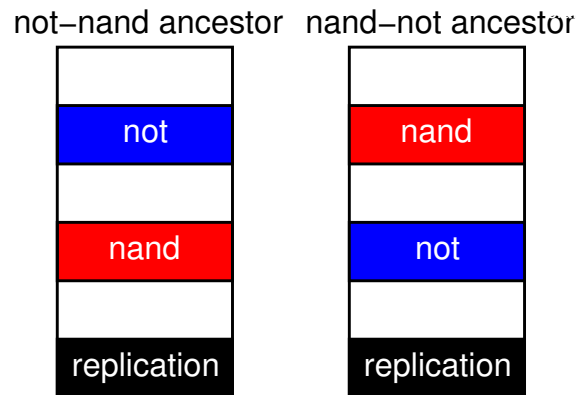
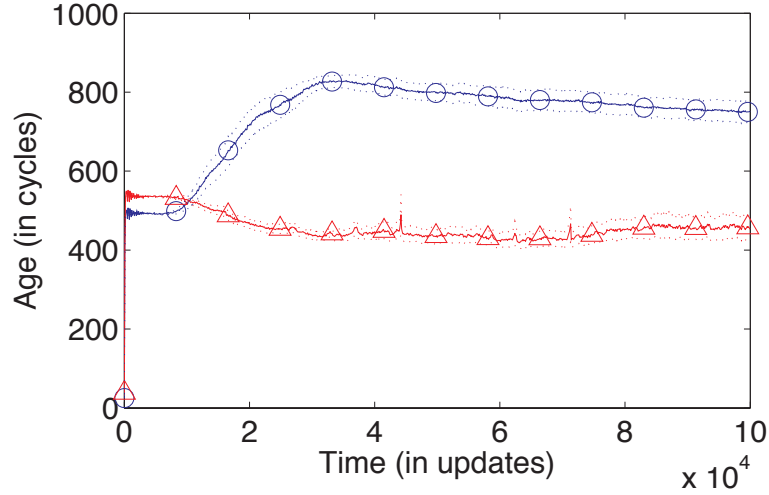


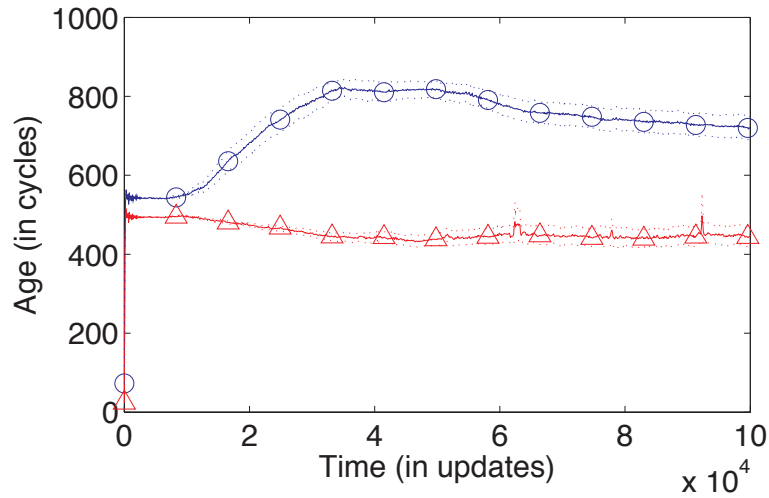
Figure 4.1: The ancestor organisms for two-task temporal polyethism experiments. The **not-nand** ancestor performs task **not**, performs task **nand**, and then replicates. The **nand-not** ancestor performs task **nand**, performs task **not**, and then replicates. Because the genomes are circular, after each organism replicates it resumes execution at the top of its genome.

Figures 4.2 and 4.3 depict the results of the experimental treatments. Within Figure 4.2, task **not** is risky. In both treatments, the mean age at which **not** is performed is significantly greater than the mean age at which **nand** is performed. For example, for the **not-nand** ancestor, **not** is performed at the mean age of 750.3773 ± 27.4510 cycles and **nand** is performed at the mean age of 453.4314 ± 29.1226 cycles. This behavior is especially interesting for the treatment that was seeded with the **not-nand** ancestor, since it required the organisms to reverse the order in which the tasks were performed. In Figure 4.3, task **nand** is risky. For both treatments, the mean age at which **nand** is performed is significantly greater than the mean age at which **not** is performed. These treatments support our hypothesis that task riskiness can result in temporal polyethism in which the more risky task is performed later in the lifetime of the organisms.

Figures 4.4 and 4.5 depict the results of our controls, which are designed to verify that, given the same level of risk, there is nothing inherent in the tasks that results in one being performed earlier or later in the organisms' lifetimes. Figure 4.4 depicts the results of the



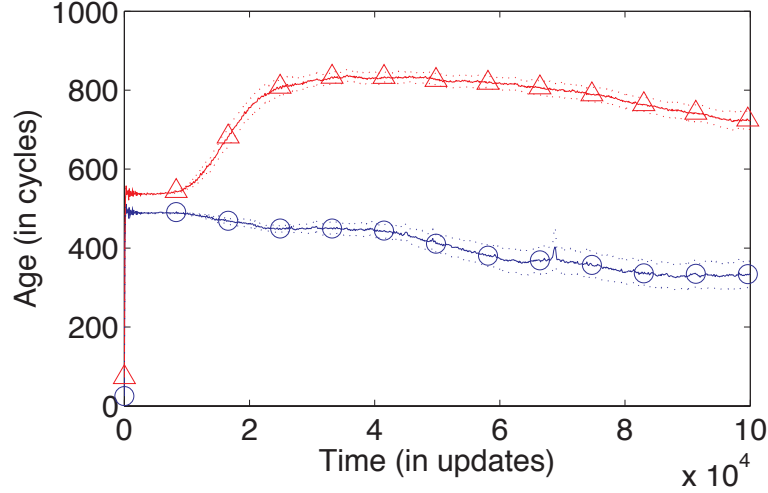
(a) Ancestor: **not-nand**; Treatment: **not is risky**



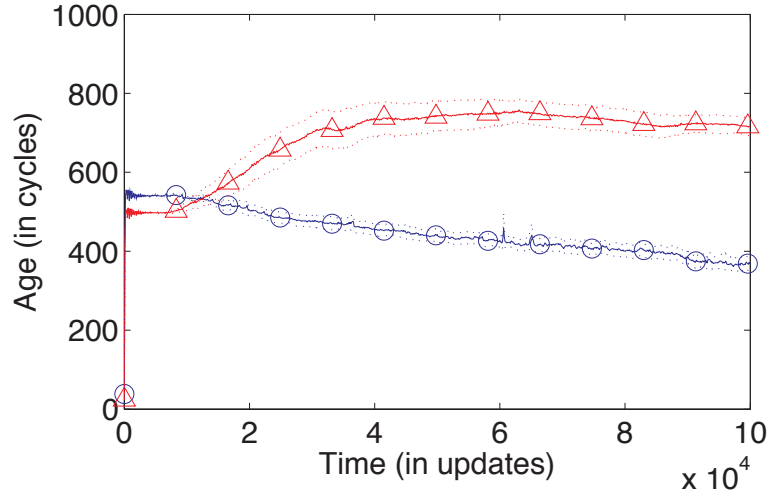
(b) Ancestor: **nand-not**; Treatment: **not is risky**

Figure 4.2: The results of the temporal polyethism treatments in which task **not** is risky. The x-axis of both plots is evolutionary time and the y-axis is the mean age in cycles. The blue line with circles represents the mean age at which task **not** is performed. The red line with triangles represents the mean age at which task **nand** is performed. In these results, the more risky task is performed later in the lifetime of the organism, which is evidence for our hypothesis that task riskiness may be an evolutionary pressure that gives rise to a temporal polyethism pattern.

control in which neither task is risky. For this control, the average age at which organisms perform tasks increases over the duration of the experiment. This change results from individual organisms evolving to perform the same task multiple times within their lifetime



(a) Ancestor: **not-nand**; Treatment: **nand** is risky



(b) Ancestor: **nand-not**; Treatment: **nand** is risky

Figure 4.3: The results of the temporal polyethism treatments in which task **nand** is risky. The x-axis of both plots is evolutionary time and the y-axis is the mean age in cycles. The blue line with circles represents the mean age at which task **not** is performed. The red line with triangles represents the mean age at which task **nand** is performed. In these results, the more risky task is performed later in the lifetime of the organism, which is evidence for our hypothesis that task riskiness may be an evolutionary pressure that gives rise to a temporal polyethism pattern.

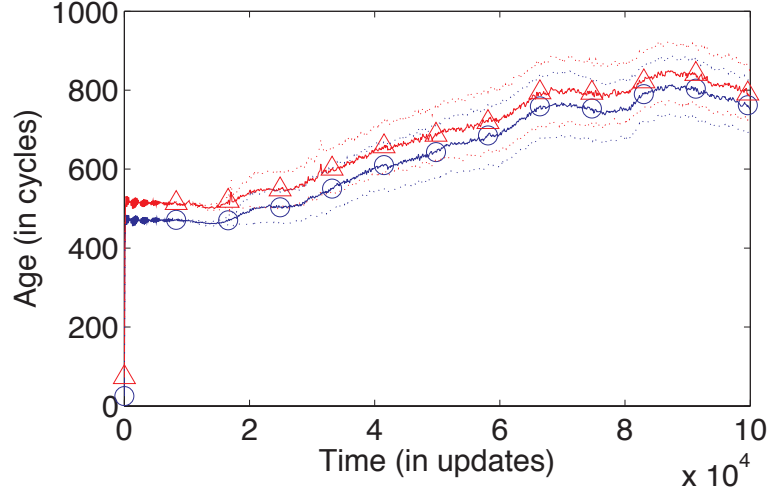
resulting in the average age of task performance increasing over the duration of the experiment. However, the mean age at which task **not** is performed is not significantly different than the mean age at which task **nand** is performed. Figure 4.5 depicts the results of the

control in which both tasks are risky. For both treatments, the mean age at which the organisms perform the tasks reflects their order in the genome. One thing to note about this control is that the high level of risk associated with both tasks decreases the rate of group replication. In fact, many groups lost the ability to replicate altogether and survived merely because other groups within their trial were also unable to replicate. Thus, these colonies are not actually evolving. However, the data provided by the controls indicate that there is nothing inherent in the **not** or **nand** tasks that implies an ordering. Taken together, these treatments indicate more risky tasks are, on average, performed later within the lifetime of the organisms.

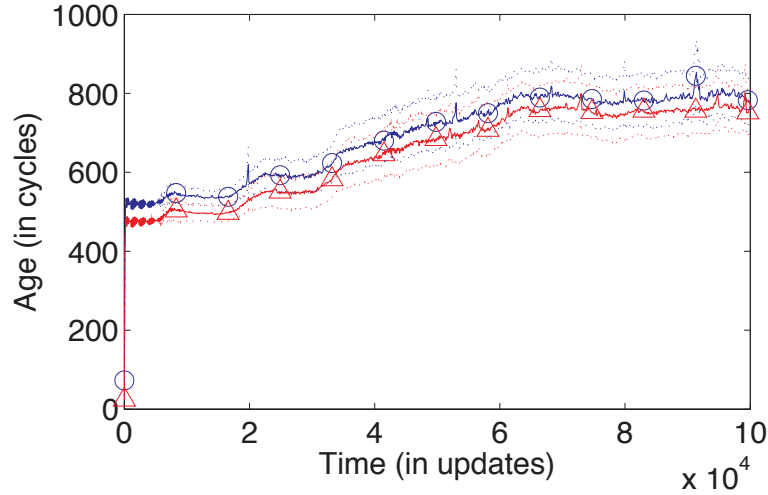
Can more complex forms of temporal polyethism result from risks associated with tasks?

In our initial temporal polyethism experiments, we created an environment in which groups were required to perform two tasks. For our next set of experiments, we created a three-task environment, in which colonies have to perform task **not** 250 times, task **nand** 250 times, and task **ornot** 250 times in order to replicate. There are three risk levels associated with the tasks – 0% lethality, 7% lethality, and 15% lethality. These risk levels add up to 22%, which is reasonably close to the 25% risk used for the previous experiment. We experimented with higher risks, however, the organisms and thus groups were prone to dying prior to replication, which resulted in the trials finishing prematurely. Using these risk levels, we created eight treatments (described in Table 4.2) that vary the riskiness associated with the tasks. The last two treatments (**g** and **h**) are controls in which either none of or all of the tasks are risky.

Additionally, we created six ancestor organisms (depicted in Figure 4.6) that all completed each task and self-replicated. These ancestor organisms were used to seed the initial colonies. Again, by varying the ancestor organism, we are able to verify that any patterns of temporal polyethism result from the riskiness associated with the tasks, not the initial



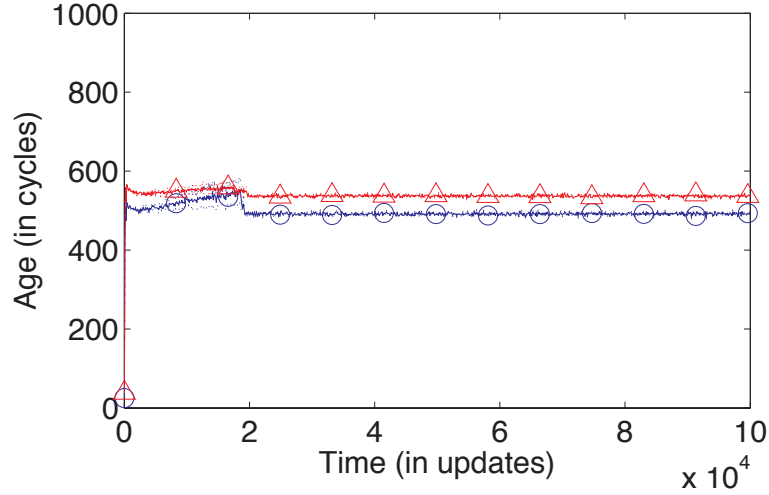
(a) Ancestor: **not-nand**; Treatment: No risk



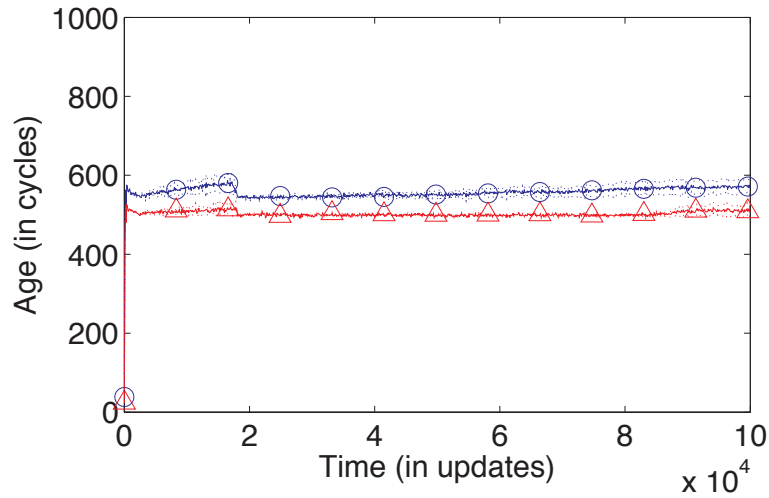
(b) Ancestor: **nand-not**; Treatment: No risk

Figure 4.4: Diagrams of the results of the control temporal polyethism treatments in which neither task is risky. The x-axis of both plots is evolutionary time and the y-axis is the mean age in cycles. The blue line with circles represents the mean age at which task **not** is performed. The red line with triangles represents the mean age at which task **nand** is performed. In these results, the controls indicate that there is nothing intrinsic about the tasks that is driving the temporal polyethism results.

genomic structure of the organisms. For each ancestor, we performed all eight treatments (described in Table 4.2) that vary the risk associated with the three tasks. As a result, we performed 1,440 trials. Our hypothesis for these experiments remains the same: task riskiness is a sufficient pressure to result in temporal polyethism regardless of which of the



(a) Ancestor: **not-nand**; Treatment: All risky



(b) Ancestor: **nand-not**; Treatment: All risky

Figure 4.5: Diagrams of the results of the control temporal polyethism treatments in which both tasks are risky. The x-axis of both plots is evolutionary time and the y-axis is the mean age in cycles. The blue line with circles represents the mean age at which task **not** is performed. The red line with triangles represents the mean age at which task **nand** is performed. In these results, the controls indicate that there is nothing intrinsic about the tasks that is driving the temporal polyethism results.

tasks is more risky or the ancestor organism. However, we expect that evolving groups that exhibit temporal polyethism should be more challenging in this environment because the differences between the risk levels of the tasks is smaller and thus selection may be weaker.

Figures 4.7-4.10 depict the results of the treatments for the **not-nand-or-not** ancestor. The

Task	Risk Treatments							
	a	b	c	d	e	f	g	h
not	0%	0%	7%	7%	15%	15%	0%	7%
nand	7%	15%	0%	15%	0%	7%	0%	7%
ornot	15%	7%	15%	0%	7%	0%	0%	7%

Table 4.2: The eight risk treatments for a three-task environment. The rows describe the lethality risks associated with tasks **not**, **nand**, and **ornot**. The columns describe a specific treatment.

results from the other experiments that use different ancestors were qualitatively similar and thus the results are not displayed. For the treatments in which the risk associated with the tasks varied (Figures 4.7-4.9), the ordering of the tasks at the end of the treatment supports our hypothesis regarding temporal polyethism. In other words, the tasks are performed in order from least to most risky. For example, in treatment Figures 4.7 **b**, task **not** is the least risky and is performed first, task **ornot** is associated with a moderate degree of risk (7%) and is performed second, and task **nand** is the most risky (15%) and is performed last. While some of these results are statistically significant, others are not. This reduction in significance may be the result of a decrease in variance in risk among tasks that, in turn, weakens selection. In general, however, these treatments indicate that the division of labor pattern of task polyethism can result from the differences in risks associated with performing tasks.

4.1.3 Analyses

We have demonstrated that groups evolve to perform more risky tasks, on average, later within their lifetime than safe tasks. Next, we examine how this behavior interacts with reproduction and then conduct two detailed analyses of groups that exhibit this behavior.

Within these experiments, organisms have a pressure not just to perform tasks, but also to replicate and produce clones capable of performing these same tasks. One question we were interested in exploring is when did the organisms replicate? To address this question,

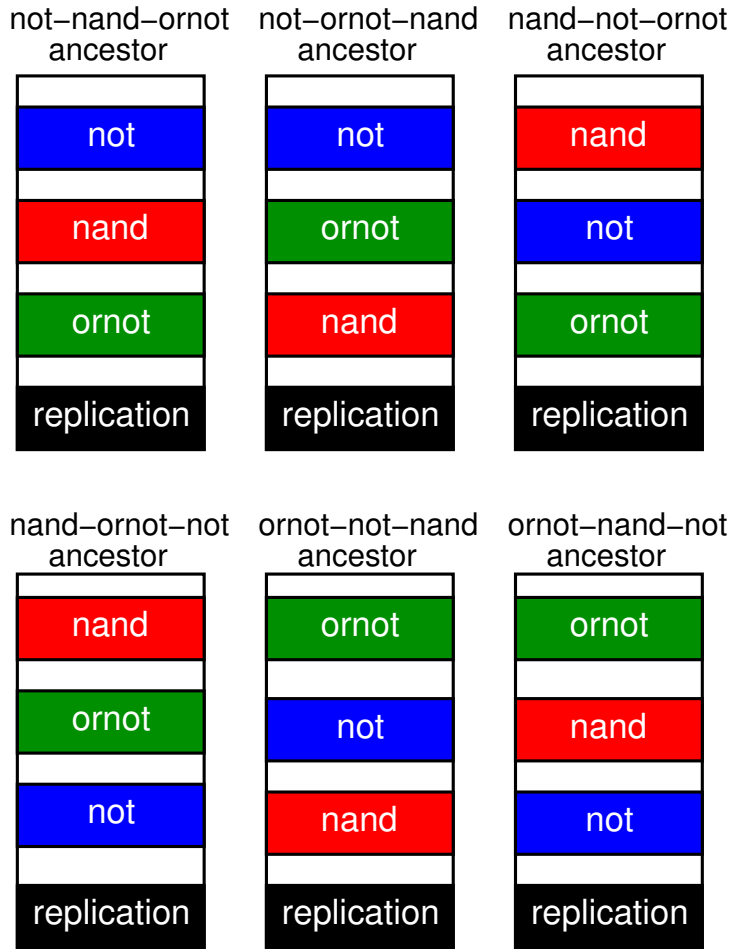
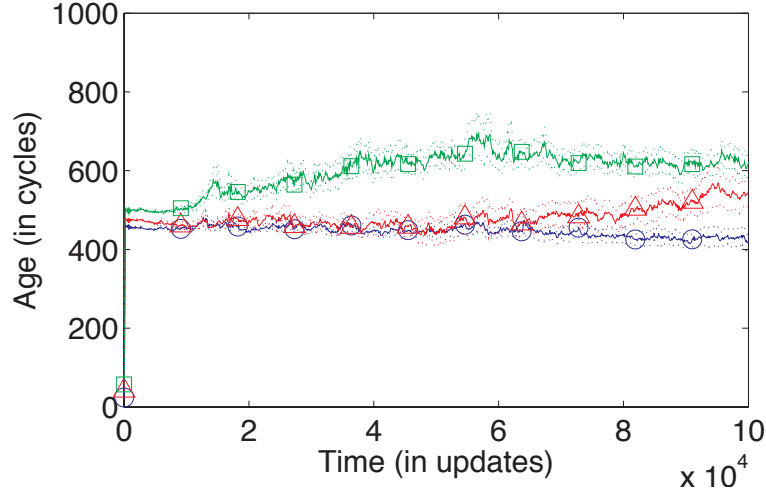
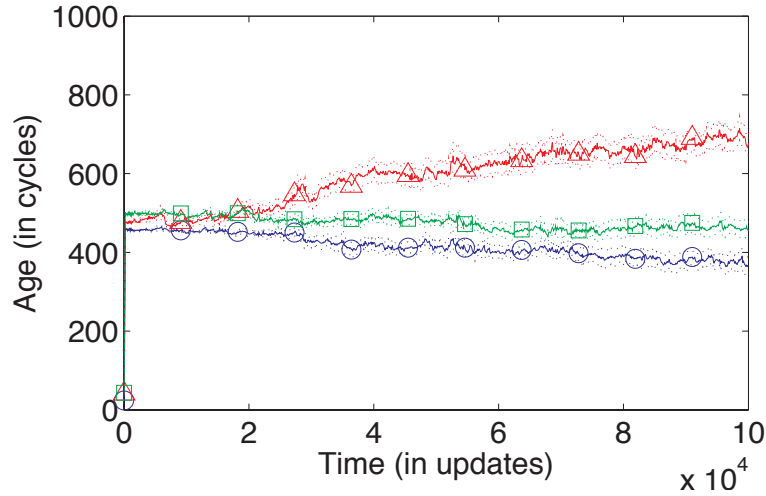


Figure 4.6: The six ancestor organisms for the three-task temporal polyethism experiments. These ancestor organisms vary the order in which the three tasks are performed. Because the genomes are circular, after each organism replicates, it resumes execution at the top of its genome.

we examined a treatment from our original two-task experiment that begins with the **not–nand** ancestor and in which task **not** is risky. Figure 4.11 depicts the mean age at which the tasks were performed and at which the organisms replicated. Intriguingly, the organisms performed the less risky task (**nand**), replicated, and then much later in their life performed the more risky task (**not**). This result suggests that the organisms have evolved a strategy that balances their need to perform tasks, the risk associated with these tasks, and their need to replicate.



(a) Lethality Risk: not 0%, nand 7%, ornot 15%

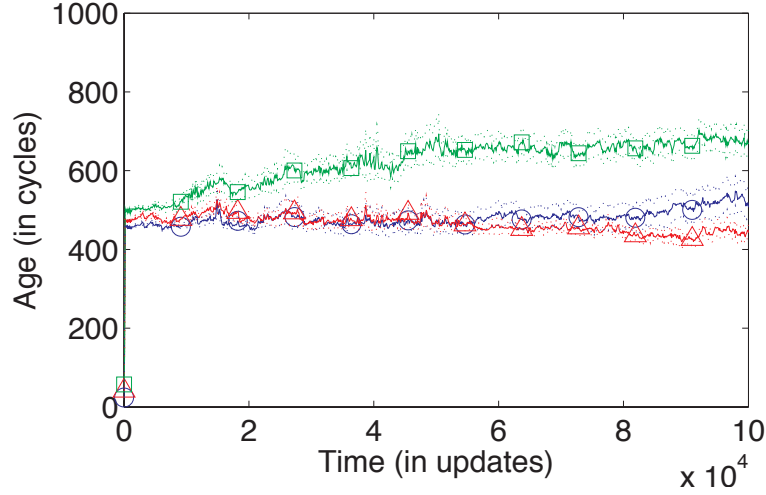


(b) Lethality Risk: not 0%, nand 15%, ornot 7%

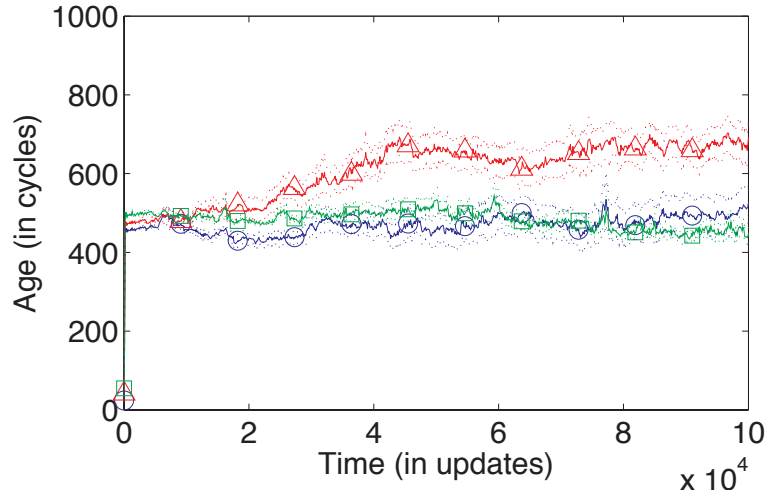
Figure 4.7: The results of the temporal polyethism experimental treatments for the not-nand-ornot ancestor in which task **not** is safe. The blue line with circles represents task **not**, the red line with triangles represents task **nand**, and the green line with squares represents task **ornot**. The x-axis is evolutionary time and the y-axis is the mean age at which the task is performed by the organisms. At the end of the treatments, the least risky task (0%) is performed first, the somewhat risky task (7%) is performed next, and the most risky task (15%) is performed last.

Two-Task Group Analysis

Next, we examined the behavior of a successful group from our original two-task experiment that begins with the **not-nand** ancestor and in which task **not** is risky to ascertain how it



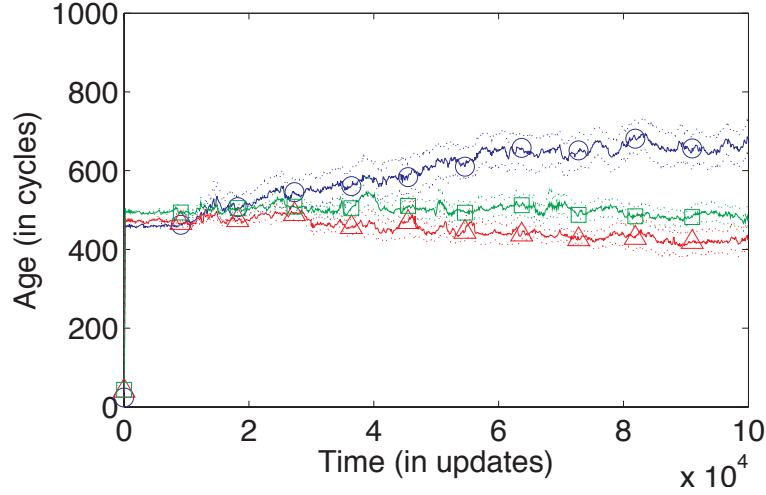
(a) Lethality Risk: not 7%, nand 0%, ornot 15%



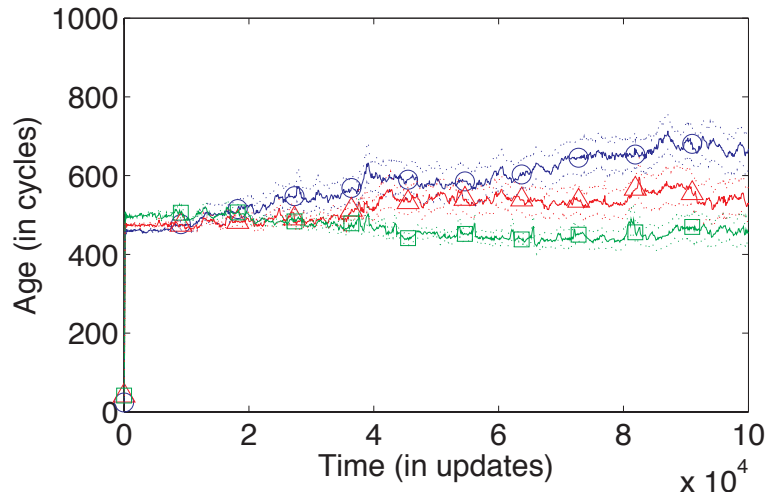
(b) Lethality Risk: not 7%, nand 15%, ornot 0%

Figure 4.8: The results of the temporal polyethism experimental treatments for the not-nand-ornot ancestor in which **not** is moderately risky. The blue line with circles represents task **not**, the red line with triangles represents task **nand**, and the green line with squares represents task **ornot**. The x-axis is evolutionary time and the y-axis is the mean age at which the task is performed by the organisms. At the end of the treatments, the least risky task (0%) is performed first, the somewhat risky task (7%) is performed next, and the most risky task (15%) is performed last.

managed task performance and replication. The organisms within this group executed a precise behavioral plan that is depicted in the phenotype portion of Figure 4.12. They performed task **nand**, replicated, performed task **nand** again, replicated again, and then



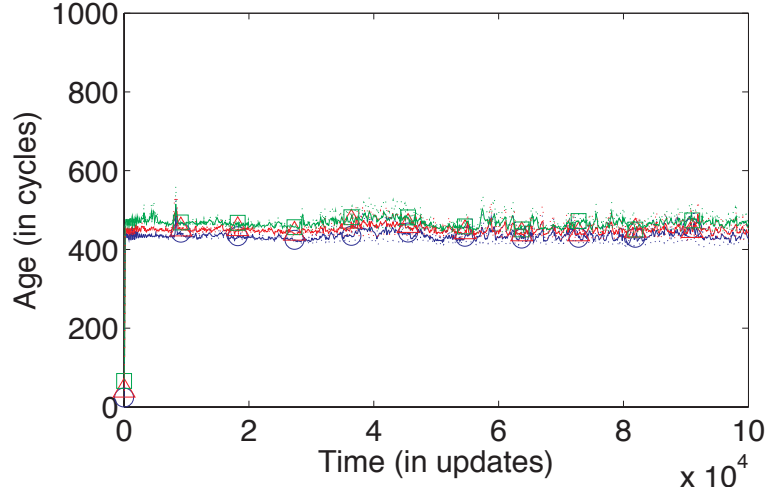
(a) Lethality Risk: not 15%, nand 0%, ornot 7%



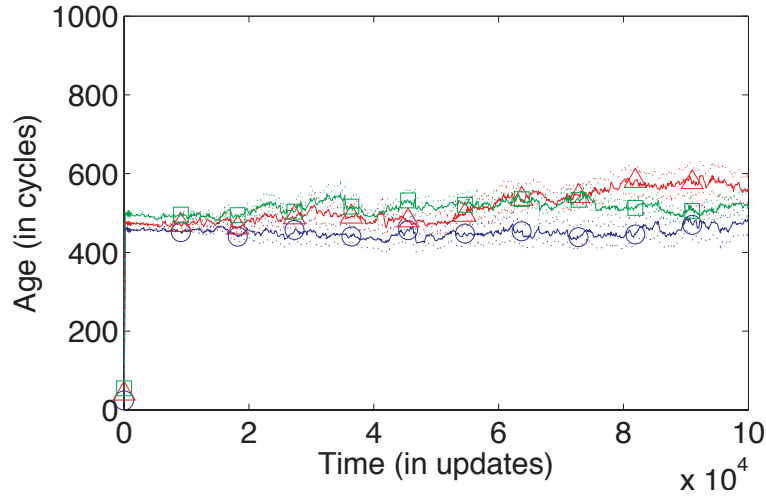
(b) Lethality Risk: not 15%, nand 7%, ornot 0%

Figure 4.9: The results of the temporal polyethism experimental treatments for the not-nand-ornot ancestor in which **not** is extremely risky . The blue line with circles represents task **not**, the red line with triangles represents task **nand**, and the green line with squares represents task **ornot**. The x-axis is evolutionary time and the y-axis is the mean age at which the task is performed by the organisms. At the end of all of the treatments, the least risky task (0%) is performed first, the somewhat risky task (7%) is performed next, and the most risky task (15%) is performed last.

repeatedly performed task **not** (the risky task) until it killed them. The organisms clearly exhibit the temporal polyethism pattern of performing the risky task after their other duties had been completed.



(a) Lethality Risk: not 0%, nand 0%, ornot 0%



(b) Lethality Risk: not 7%, nand 7%, ornot 7%

Figure 4.10: The results of the two temporal polyethism control treatments for the not-nand-ornot ancestor. The blue line with circles represents task **not**, the red line with triangles represents task **nand**, and the green line with squares represents task **ornot**. The x-axis is evolutionary time and the y-axis is the mean age at which the task is performed by the organisms. The ages at which the tasks are performed are not significantly different.

A second topic we are interested in exploring is how the genome was modified to support this behavior. For example, organisms may have rearranged their genome to support task ordering (i.e., by moving the more risky task to the end of their genome) or organisms may have evolved to use control-flow instructions that enable them to skip over portions

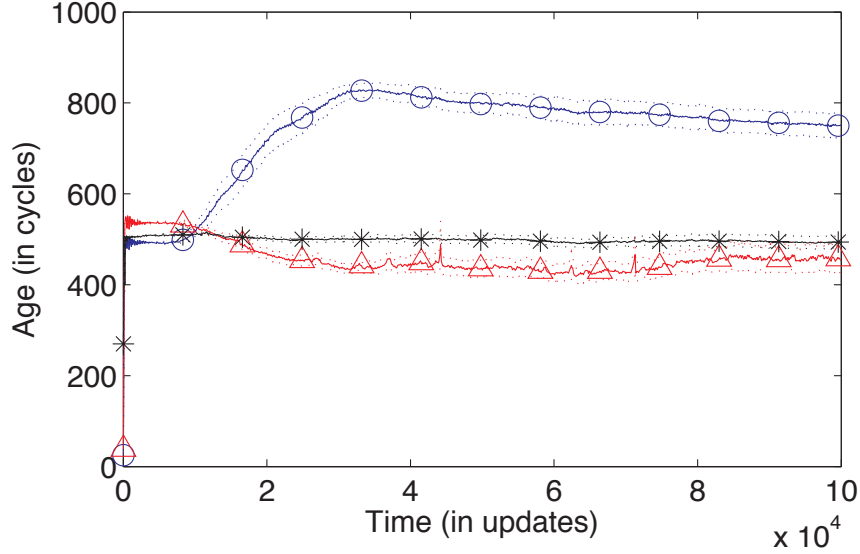


Figure 4.11: These results depict the mean age at which task **not** (blue line with circles), task **nand** (red line with triangles) and replication (black line with stars) are performed for the treatment where **not** is risky and the runs were started with the **not-nand** ancestor. These results suggest that the organisms are performing task **nand** one or more times, replicating, and then performing task **not**.

of their genome. In this case, the organisms evolved to use the control-flow instructions. The architecture of the genome, which is depicted in the genotype portion of Figure 4.12, is extremely similar to the ancestor organism: task **not** is encoded first, then task **nand**, and lastly replication. However, the organisms evolved in both jump instructions (to skip task **not** until the remainder of the genome had been executed twice) and a loop to continue to perform task **not** until death. Organisms set and used the value of a register that was preserved during replication to track which genome iteration they were on and to modify their behavior accordingly. The numbered arrows in Figure 4.12 depict the order in which the elements of the genome were executed.

Three-Task Group Analysis

Next, we examined the behavior of a group that appears to exhibit temporal polyethism within the three-task environment. Specifically, we examined the behavior of a group that

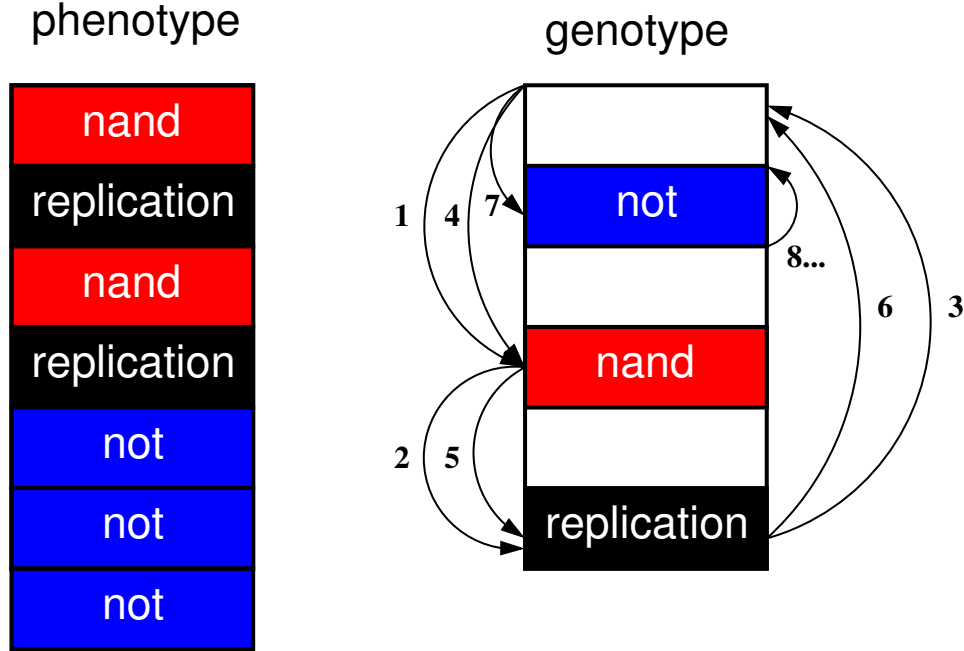


Figure 4.12: Diagrams of the phenotype (left) and genotype (right) of an organism whose group exhibited temporal polyethism with two tasks. The numbered arrows surrounding the genotype indicate the order in which instructions were executed to produce the phenotype. In this case, the genotype is very similar to the **not-nand** ancestor. The risk-based order in which the tasks were performed depended upon control-flow instructions in the genome.

evolved from the **not-nand-ornot** ancestor and that evolved under risk treatment **c** (Table 4.2 and Figure 4.8), where **nand** was safe, **not** incurred moderate (7%) risk, and **ornot** incurred a high degree of risk (15%). We seeded a test group with the genotype of the group and monitored it to better understand the organisms' behavior. The organisms within this group executed a complicated behavioral plan that is depicted in the phenotype portion of Figure 4.13. When an organism finished executing this pattern, it started again. These organisms exhibit almost perfect age-based castes; task **nand** is performed first, then task **not**, and lastly task **ornot**.

When we examined the genotype of the organisms we noted that their genetic structure differed from the ancestor organism from which they evolved. The ancestor organism performed task **not**, then **nand**, and lastly **ornot** order. In contrast, the evolved organism re-

arranged its genome (depicted in Figure 4.13) to encode task **nand**, task **not**, task **not** again, and then task **ornot**. In addition to the genomic modifications, the genotype also made use of control-flow instructions to achieve temporal polyethism.

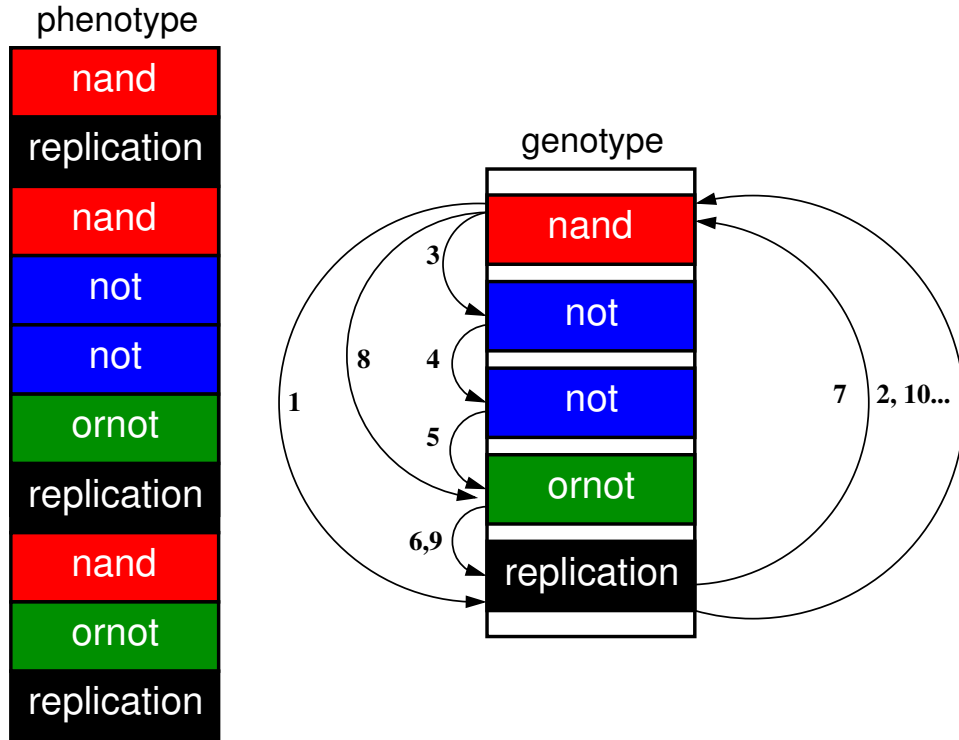


Figure 4.13: Diagrams of the phenotype (left) and genotype (right) of an organism whose group exhibited temporal polyethism with three tasks. The numbered arrows surrounding the genotype indicate the order in which instructions were executed to produce the phenotype. In this case, the genotype of the **not-nand-ornot** ancestor has been rearranged to facilitate temporal polyethism. Specifically, **not** and **nand** have exchanged places and an additional **not** has been introduced. The risk-based order in which the tasks were performed depended upon both the genomic architecture and the control-flow instructions in the genome.

4.1.4 Discussion

In this section, we described how we have used Avida to explore one hypothesis for the origin of temporal polyethism division of labor. Specifically, we found support for the age polyethism hypothesis, where the age of an individual predicts the task it specializes on. This

age polyethism structure enables the group to be more efficient at gathering resources. In our analyses, we found further evidence that organisms made use of control flow instructions and genomic architecture modifications to achieve this behavior.

4.2 Task-Switching Costs

As previously mentioned, three advantages for human economic division of labor suggested by Adam Smith [108] and later proposed by Dornhaus [20] as possible reasons for division of labor within eusocial colonies are: (1) the efficiency with which an individual performs a task improves with repeated performances, (2) specialization can decrease task-switching costs, and (3) the creation of machines can assist organisms performing specialized tasks more efficiently. In this study, we examine how task-switching costs (the second proposed benefit) affect whether or not division of labor evolves as a strategy [41, 44]. Within natural systems, task-switching costs can have many separate components including the physical time necessary to move between the locations for different tasks, the cognitive overhead in changing tasks, and morphological overhead in changing roles. For example, if a forager honeybee changes to become a nurse honeybee, then it must move from the dance floor located at the bottom of the hive to the top of the hive where the brood live and make morphological gland adjustments to be able to feed the brood. These various task-switching costs present within nature are difficult to identify and challenging to experimentally manipulate. In this study, we use Avida to address whether task-switching costs present in the environment affect whether division of labor strategies evolve within colonies.

4.2.1 Methods

To use Avida to study the evolution of division of labor, we created a world consisting of competing groups that represent colonies. These groups comprise clonal organisms that are

rewarded for their overall group performance. Once the group as a whole collects a number of resources, then the group is rewarded by reproducing using germ-line replication. The speed at which a group is able to collect resources, and thus the overall performance of the group, is improved if the individuals within the group perform a variety of different types of tasks. This dynamic is analogous to bee colonies whose performance is improved if they collect both nectar and pollen [89]. For our initial experiments, we configured nine limited resources – one for each of the logic tasks. Each resource had 100 initial units with an inflow rate of 1 unit per update and an outflow rate of 1% of the total resource per update. When an organism performs a task, it amasses up to 5% of the available resources associated with that type of task.

A group replicates when it has amassed 500 units of resources. Given these parameters, the most efficient way for a group to replicate is to perform each type of task. By contrast, if a group performs only a single type of task, it takes approximately 600 updates to replicate. Notably, organisms within the groups can accomplish this objective either as generalists or as specialists. A generalist organism may perform each type of task once; whereas, a specialist organism may perform a single type of task many times relying on the other organisms in the group to perform the other types of tasks. To start these experiments, we seeded all groups with one organism that performs one logic task (**not**). Mutations occur during group replication, potentially enabling a group to perform an additional type of task. Over time, groups may evolve to perform a diversity of types of tasks and potentially even strategies that include coordinating their task performance and thus performing division of labor.

To address our central question of the effect of task-switching costs on the evolution of division of labor, we extended Avida to support a performance penalty that was applied each time an organism changed the type of task it was performing. We then created three separate treatments: (1) a control in which task-switching costs were not present, (2) a moderate task-switching-cost environment in which a small performance penalty was incurred each

time an organism changed the type of task it performed, and (3) a high task-switching-cost environment in which a large performance penalty was incurred each time an organism changed the type of task it performed. We use Shannon mutual information as proposed by Gorelick *et al.* [48] to measure the amount of division of labor present within the groups.

For each experiment, we conducted 50 trials to account for the stochastic nature of evolution. Within each trial, we used the default group topology with genetically-clonal digital organisms. All germlines are fixed at a length of 100 instructions. Mutations occur to the germline when the group replicates; the mutation rate is set at 1.0 genomic. The trials last for 201,000 updates. After the first 200,000 updates, the groups go through a 1000 update ecological period, where the mutation rate is set to 0. The ecological period prunes spurious mutations and enables us to better analyze and assess the behavior of the groups.

4.2.2 Results

We examine the effects of task-switching costs on division of labor by addressing the following three questions:

- Do task-switching costs result in an increase in the evolution of division of labor?
- Does the type of tasks that organisms must perform affect the adoption of division of labor as a strategy?
- Does the amount of resources required for a group to replicate affect the adoption of division of labor as a strategy?

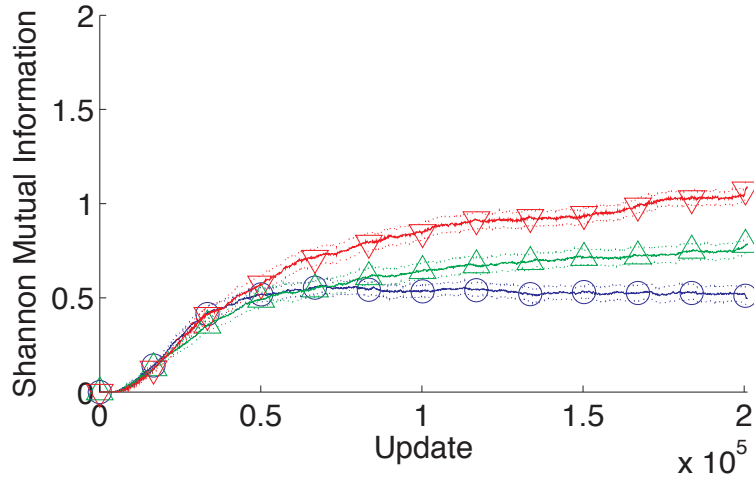
Do task-switching costs result in an increase in the evolution of division of labor?

The central question we are investigating is whether division of labor will arise more frequently as a strategy when groups are rewarded for overall performance and there is a cost to switching tasks. As part of this experiment, we vary task-switching costs (measured in

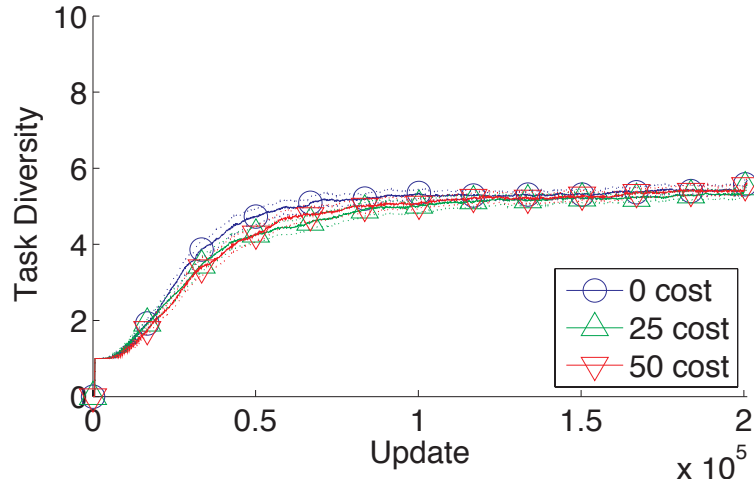
terms of extra cycles) while holding other other experimental parameters constant. Specifically, we experiment with costs of 0 (a control), 25, and 50 cycles. In all cases, a group that performs multiple types of tasks will outcompete a group the performs only a single task. The question is whether these tasks are performed by generalist or specialist organisms. Without task-switching penalties, a successful group could comprise generalist individuals that each perform each type of task once. However, with task-switching penalties, a more efficient group would comprise specialist organisms that perform just one type of task.

Figure 4.14(a) depicts how the Shannon mutual information between individuals and tasks (our measurement of division of labor) changes over the course of evolutionary time for the three task-switching-cost treatments. The blue line with circles is the control; the green line with triangles is the 25-cost treatment; the red line with upside-down triangles is the 50-cost treatment. These data support our hypothesis that as task-switching costs increase, there is an increase in division of labor, which is indicated by a rise in Shannon mutual information between organisms and tasks performed (0-cost, 0.494 ± 0.043 ; 25-cost, 0.788 ± 0.039 ; 50-cost, 1.086 ± 0.041). All three treatments are significantly different from one another (50-cost compared to 0-cost: $p=1.5849e-13$, 25-cost compared to 0-cost: $p=4.6288e-06$, and 50-cost compared to 25-cost: $p=4.6288e-06$; all p-values presented in this section were computed using Wilcoxon rank sum test).

Figure 4.14(b) depicts the number of different types of tasks performed by the successful groups at replication for the three task-switching cost treatments. These data indicate that all of the groups for this environment are performing approximately 5-6 types of tasks and that the number of types of tasks performed does not vary with the task-switching cost applied. Thus, differences in the amount of division of labor present within the groups reflects the underlying level of specialization of members of the groups and is not a side-effect of the number of types of tasks performed. Additionally, these data explain the initial increase in the amount of Shannon mutual information for the 0-cost treatment: as the



(a) Mean Shannon Mutual Information



(b) Mean Number of Unique Tasks Performed by Replicating Groups

Figure 4.14: Logic-9 environment results: (a) The mean Shannon mutual information between organisms and tasks performed averaged across 50 runs for groups with varying amounts of task-switching costs. Dotted lines are used to indicate standard error. Notably, treatments with higher task-switching costs evolve strategies that exhibit higher levels of division of labor. (b) The mean number of different types of tasks performed by the groups under various treatments. The groups typically evolve to perform 5-6 tasks.

organisms evolved to perform more types of tasks, their phenotypes differed as a result of the timing of death and replication and thus Shannon mutual information rose.

These results indicate that an environment that imposes task-switching costs is a sufficient

pressure to give rise to groups that employ division of labor strategies. Additionally, this increase can be attributed to organisms within the groups becoming increasingly specialized as task-switching costs increase. Our next two sets of experiments test the robustness of these results by varying different aspects of the environment.

Do the tasks in the environment affect the adoption of division of labor as a strategy?

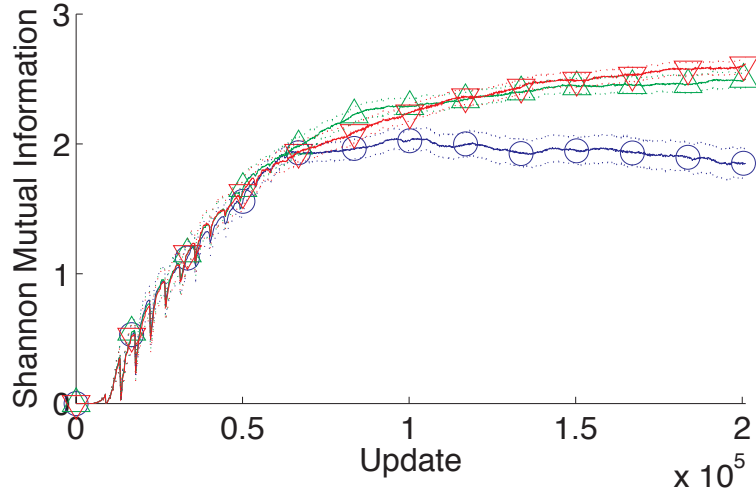
Our central experiment suggested that task-switching costs can lead to the evolution of strategies that employ division of labor. Next, we explore whether the type of task present in the environment affects whether or not division of labor evolves. To study this question, we returned to the 25 role-selecting tasks. We associate each role-selecting task with a limited resource that has an initial amount of 100 resources, an inflow of 0.25 resources per update, and an outflow of 1% per update. When an organism performs a task, it can consume up to 5% of the available resources associated with that type of task. Because the groups have more resources at their disposal, we also increase the amount of resources required for replication to 1000 resources. We ran 50 replicates of each cost treatment (i.e., 0-cost, 25-cost, and 50-cost). All groups were seeded with an individual that set its role-id to one.

Figure 4.15 (a) depicts the results of the various treatments in the 25-role environment. At the final time point in the treatment, the mean amount of Shannon mutual information between organisms and tasks performed is: 0-cost: 1.8467 ± 0.1163 ; 25-cost: 2.503 ± 0.0638 ; 50-cost: 2.6145 ± 0.0494 . In this case, for all treatments, the groups are performing approximately 20 different roles (Figure 4.15(b)). As a result, we can conclude that differences in division of labor result solely from organisms choosing to be generalists and specialists. For this experiment, we see more division of labor is present in runs with task-switching costs, than the control runs (50-cost to 0-cost: $p=9.6939e-08$; 25-cost to 0-cost: $p=7.5509e-06$),

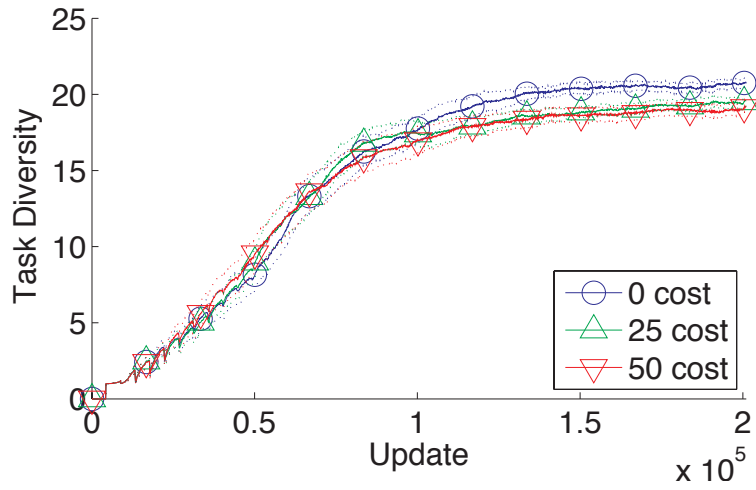
which supports our hypothesis that task-switching costs increase the amount of division of labor present in evolved strategies. However, the 50-cost treatment is not significantly different than the 25-cost treatment ($p=0.1777$). This result may be due to the 25-cost treatments already achieving near perfect division of labor scores.

To further explore this question, we created an environment that combined the logic tasks and the role-selecting tasks. Specifically, we included all nine logic tasks and six of the role-selecting tasks (i.e., roles 1-6) for a total of 15 tasks rewarded in the environment. Each task has an associated limited resource with an initial amount of 100 resources, an inflow of 0.6 resources per update, and an outflow of 1% per update. When an organism performs a task, it can consume up to 5% of the available resources associated with that type of task. Because the groups have more resources at their disposal, we set the amount of resources required for replication to 1000 resources. We ran 50 replicates of each cost treatment (i.e., 0-cost, 25-cost, and 50-cost). All groups were seeded with an individual that set its role-id to one.

Figure 4.16 (a) depicts the results of the various treatments in the new logic-9-role-6 environment. At the final time point in the treatment, the mean Shannon mutual information between organisms and tasks performed is: 0-cost: 1.3876 ± 0.0352 ; 25-cost: 1.6596 ± 0.0220 ; 50-cost: 1.7158 ± 0.0148 . In this case, for all treatments, the groups are performing approximately eight to eleven different types of tasks (Figure 4.16(b)). However, these differences are not statistically significant. As a result, we can again conclude that differences in division of labor result solely from organisms choosing to be generalists and specialists. For this experiment, we see more division of labor is present in runs with task-switching costs, than the control runs (50-cost to 0-cost: $p=9.4320e-13$; 25-cost to 0-cost: $p=3.8440e-09$), which supports our hypothesis that task-switching costs increase the amount of division of labor present in evolved strategies. However, the 50-cost treatment is not significantly different than the 25-cost treatment ($p=0.1845$).



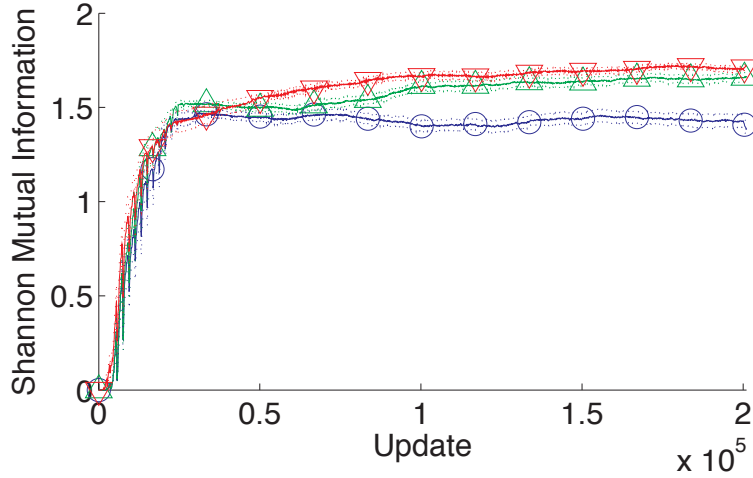
(a) Mean Shannon Mutual Information



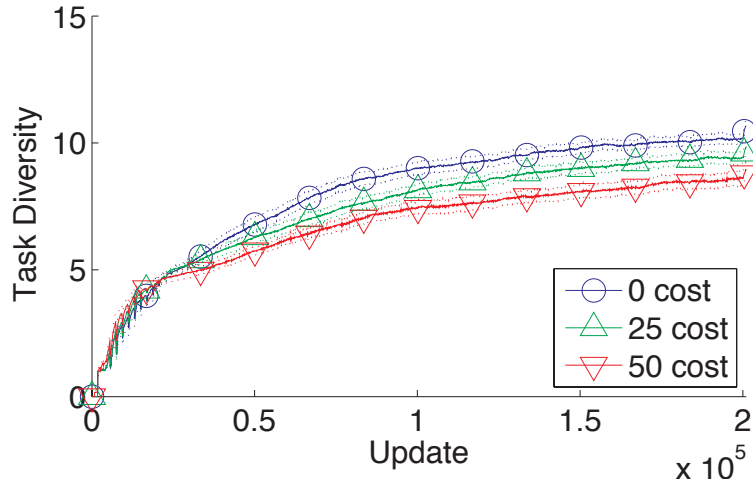
(b) Mean Number of Unique Tasks Performed by Replicating Groups

Figure 4.15: 25-Role Environment Results: (a) The mean Shannon mutual information between organisms and tasks performed averaged across 50 runs for groups with varying amounts of task-switching costs within the 25-role environment. Dotted lines are used to indicate standard error. Notably, treatments with higher task switching costs evolve strategies that exhibit higher levels of division of labor. (b) The mean number of different types of tasks performed by the groups under various treatments. The groups all evolve to perform approximately 20 types of tasks.

In general, these results support our central hypothesis that task-switching costs promote the evolution of division of labor. Next, we return to the logic-9 environment and test a different environmental parameter.



(a) Mean Shannon Mutual Information



(b) Mean Number of Unique Tasks Performed by Replicating Groups

Figure 4.16: Logic-9 and Role-6 Environment Results: (a) The mean Shannon mutual information between organisms and tasks performed averaged across 50 runs for groups with varying amounts of task-switching costs. Dotted lines are used to indicate standard error. Notably, treatments with higher task switching costs evolve strategies that exhibit higher levels of division of labor. (b) The mean number of different types of tasks performed by the groups under various treatments. The groups all evolve to perform approximately 10 types of tasks.

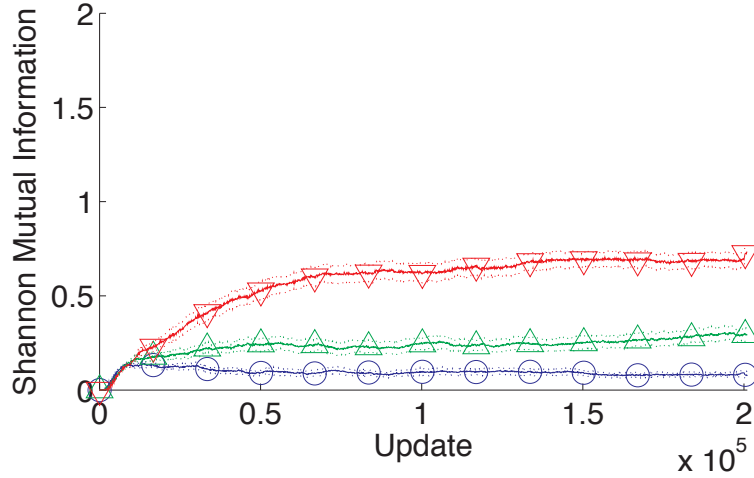
Does the amount of resources required to replicate affect the adoption of division of labor as a strategy?

In our previous experiments, we demonstrated that task-switching costs increased the amount of division of labor present in the evolved groups for two different types of environ-

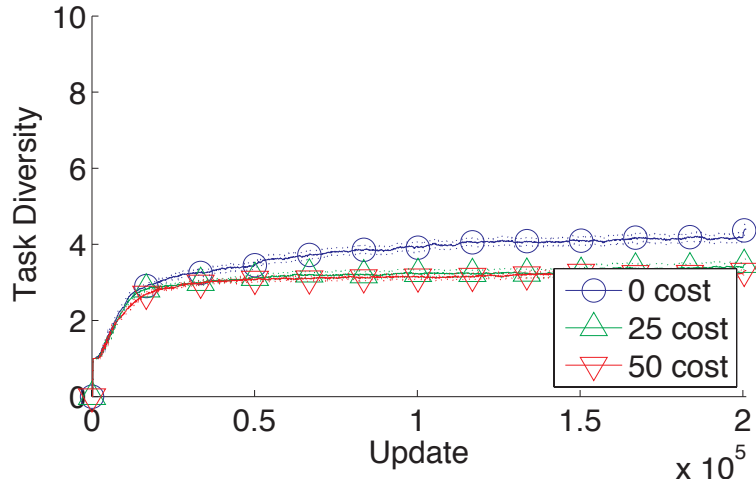
mental tasks. To ensure that these results are not an artifact of the specific environmental parameters selected, we performed additional sets of experiments in which we varied the amount of resources required for the group to replicate. Specifically, the original experiment required the groups to amass 500 resources. Our two additional experiments required groups to amass 250 (half the original amount) and 1000 resources (double the original amount) prior to replicating. For each experiment, we ran all three task-switching cost treatments.

Figure 4.17 depicts the results of the task-switching-cost treatments under the 250 resource requirement. For this experiment, we see that as task-switching costs increase, there is an increase in division of labor indicated by an increase in Shannon mutual information between organisms and tasks performed. At the final time point in the treatment, the mean Shannon mutual information is: 0-cost: 0.074 ± 0.018 ; 25-cost: 0.2913 ± 0.0380 ; 50-cost: 0.737 ± 0.0385 . All three of these treatments are statistically significantly different from one another (50-cost to 0-cost: $p=8.1570e-16$; 25-cost to 0-cost: $p=2.2937e-07$; 50-cost to 25-cost: $p=3.6761e-10$). Additionally, with this lower resource requirement, the variety of types of tasks performed by the groups decreases (Figure 4.17(b)). For these treatments, the control (i.e., 0-cost) groups performed an average of 4 types of tasks and the higher task-switching cost treatments (i.e., 25-cost and 50-cost) performed an average of 3 types of tasks. This difference in the number of types of tasks performed could potentially bias the results in favor of higher division of labor scores for the 0-cost treatment, since a greater variety of tasks has the potential to increase division of labor scores. Since this bias is in the opposite direction of our prediction, we are not concerned with the performance difference.

For the 1000 resource treatments, the results of which are depicted in Figure 4.18, it is challenging to visually ascertain if there is a significant increase in division of labor resulting from the application of task-switching costs. At the final time point in the treatment, the mean amount of Shannon mutual information between organisms and tasks performed is: 0-cost: 0.7553 ± 0.0442 ; 25-cost: 0.8698 ± 0.0459 ; 50-cost: 0.9089 ± 0.0519 . For these



(a) Mean Shannon Mutual Information



(b) Mean Number of Unique Tasks Performed by Replicating Groups

Figure 4.17: Logic-9 250 Resource Results: (a) The mean Shannon mutual information between organisms and tasks performed averaged across 50 runs for groups with varying amounts of task-switching costs. Dotted lines are used to indicate standard error. Notably, treatments with higher task-switching costs evolve strategies that exhibit higher levels of division of labor. (b) The mean number of different types of tasks performed by the groups under various treatments. The groups all evolve to perform 3-4 types of tasks.

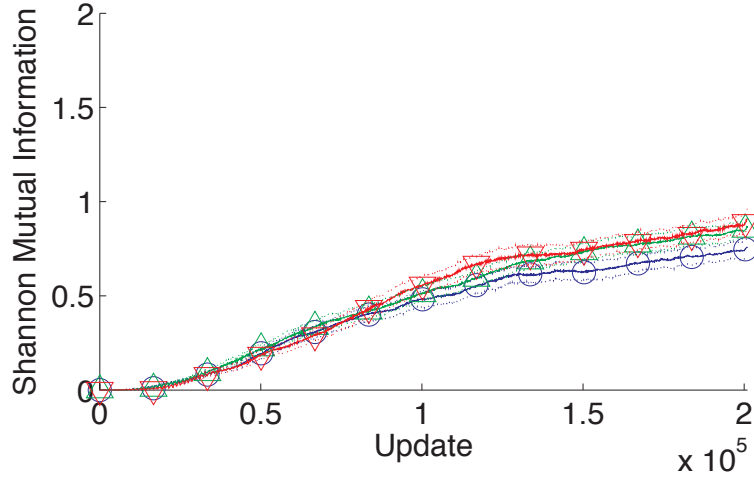
treatments, the amount of Shannon mutual information present in the 50-cost treatment is significantly greater than the 0-cost treatment ($p=0.0396$), but not significantly greater than the 25-cost treatment ($p=0.7433$); nor is the 25-cost treatment significantly different from

the 0-cost treatment ($p=0.0652$). With the increased resource requirement, the variety of types of tasks performed by the all of the treatments increases to 7-8 tasks (Figure 4.18(b)).

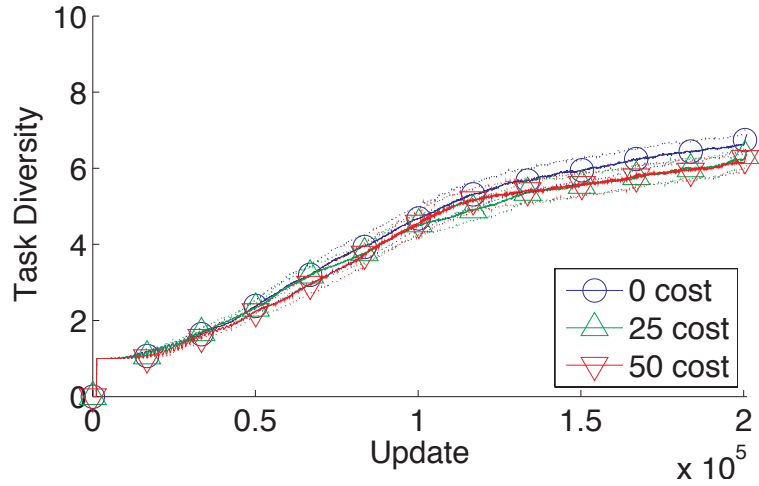
One intriguing aspect of these data is that the lack of significance results from an increase in division of labor exhibited by the 0-cost treatment. The treatment mean increased from 0.074 ± 0.018 in the 250-resource experiment to 0.7553 ± 0.0442 in the 1000-resource experiment. To understand this phenomena, we must more closely examine the effects of increasing the resources required for a group to replicate. In both the 250 and 500 resource experiments, amassing the amount of resources present at the start of the experiment (100 per task type for a total of 900 resource units) was sufficient for a group to replicate. However, in the 1000 resource experiment, there were not enough resources present in the initial environment and thus groups needed to wait for resources to flow into the environment in order to replicate. This resource dynamic resulted in a pressure for groups to amass resources as quickly as possible and thus to perform all tasks at each time point. As a result, 0-cost treatment groups used division of labor to increase their overall efficiency through parallel processing. While this was not the hypothesis we were testing for the evolution of division of labor, efficiency through parallel processing has previously been proposed as a benefit of division of labor [53, 88]. Thus, within this experiment, we have uncovered evidence for both the role of task-switching costs and efficiency through parallel processing in the evolution of division of labor.

4.2.3 Analyses

We have demonstrated that groups employ a larger amount of division of labor when they evolve in the presence of task-switching costs. Next, we examine what mechanisms the groups use to perform division of labor and then study the strategy of one group in particular.



(a) Mean Shannon Mutual Information



(b) Mean Number of Unique Tasks Performed by Replicating Groups

Figure 4.18: Logic-9 1000 Resource Results:(a) The mean Shannon mutual information averaged across 50 runs for groups with varying amounts of task-switching costs. Dotted lines are used to indicate standard error. (b) The mean number of different types of tasks performed by the groups under various treatments. The groups all evolve to perform 7-8 types of tasks.

Mechanisms used to perform division of labor.

For our experiments, we provided organisms with instructions that enable explicit coordination using messaging and location-awareness. A group could evolve a genome that uses these instructions to coordinate roles. For example, organisms could select a type of task to

perform based on their location or could send messages alerting other organisms of their selected task type. To better understand whether the successful groups used these instructions to coordinate the tasks performed by their constituent organisms, we explored the behavior of groups that successfully exhibited division of labor in the 25-role and logic-9 environment. Based on our preliminary results, we suspect that the type of task will influence the mechanism used by the group. For these analyses, we selected the best performing group from each of the 50 trials for the 50-cost treatment for both the logic-9 and the 25-role environment. We then conducted knockout experiments, where we replaced one or more instructions in the germline of a group with a neutral instruction (called **nop-X**). We reran each group stopping either when the group replicated or at 5000 updates. If the group’s ability to perform division of labor was impaired following a knockout, then we can conclude that the knocked-out instruction contributed to the collective behavior of the group. We performed two sets of knockout experiments: (1) knockouts of location sensing capabilities (i.e., the **get-cell-xy** and **collect-cell-data** instructions), (2) knockouts of messaging capabilities (i.e., the **send-msg** and **retrieve-msg** instructions).

Figures 4.19 and 4.20 contain box plots that describe how the amount of division of labor was influenced by the various knockout treatments for the logic-9 and 25-role environments, respectively. Within both of these plots, the individual boxes depict how the amount of Shannon mutual information changes as we knockout instructions. The first boxes (**control**) are the baseline performance of the group without any loss of information. The second boxes (**ko-location**) are the performance of the groups after knocking out the location-sensing instructions. The third boxes are the performance of the groups after knocking out the messaging instructions (**ko-messaging**).

These data suggest that the logic-9 groups evolved division of labor strategies that included coordinating with neighboring organisms through the use of messaging. For the control and location knockout analyses, the logic-9 groups achieved median Shannon mutual

information scores of 1.0534 and 1.07072, respectively. However, for the messaging knockout treatment, the same groups achieved a median Shannon mutual information score of 0. Conversely, the 25-role groups appear to have evolved division of labor strategies that included coordinating roles using location. For the control and messaging knockout analyses, the 25-role groups achieved median Shannon mutual information scores of 2.38828 and 2.59492, respectively. However, for the location knockout treatment, the same groups achieved a median Shannon mutual information score of 0. Examining the data, it was evident that division of labor scores plunged to 0 in both cases because their coordination capabilities played such an integral role in the strategy of the group that organisms were unable to successfully amass the necessary resources within 5000 updates and thus many of the groups were unable to replicate. These results confirm our previous results that groups are evolving strategies to coordinate their roles and, moreover, these strategies are dependent on the type of task they must perform.

Analysis of a group strategy for achieving division of labor.

To better understand the behavior of a specific group, we examined a group from the 50-cost treatment that successfully performed division of labor in the 25-role environment. This group performed 24 of 25 role-selecting tasks. Each organism within the group performed one type of task multiple times. Figure 4.21 depicts the role-id selected by each member of the group.

To understand how organisms selected these roles, we examined the genome of the group. Figure 4.22 depicts an excerpt of the genome. This genome loads the x and y coordinate of the organism into the **AX** and **BX** registers. It then adds the contents of the **BX** register to the **AX** register five times and then sets the role-id of the organism to be the contents of the **AX** register. Essentially, this algorithm sets the role-id of the organism to be the sum of the column-id and five times the row-id. For example, an organism at position (3, 2), would

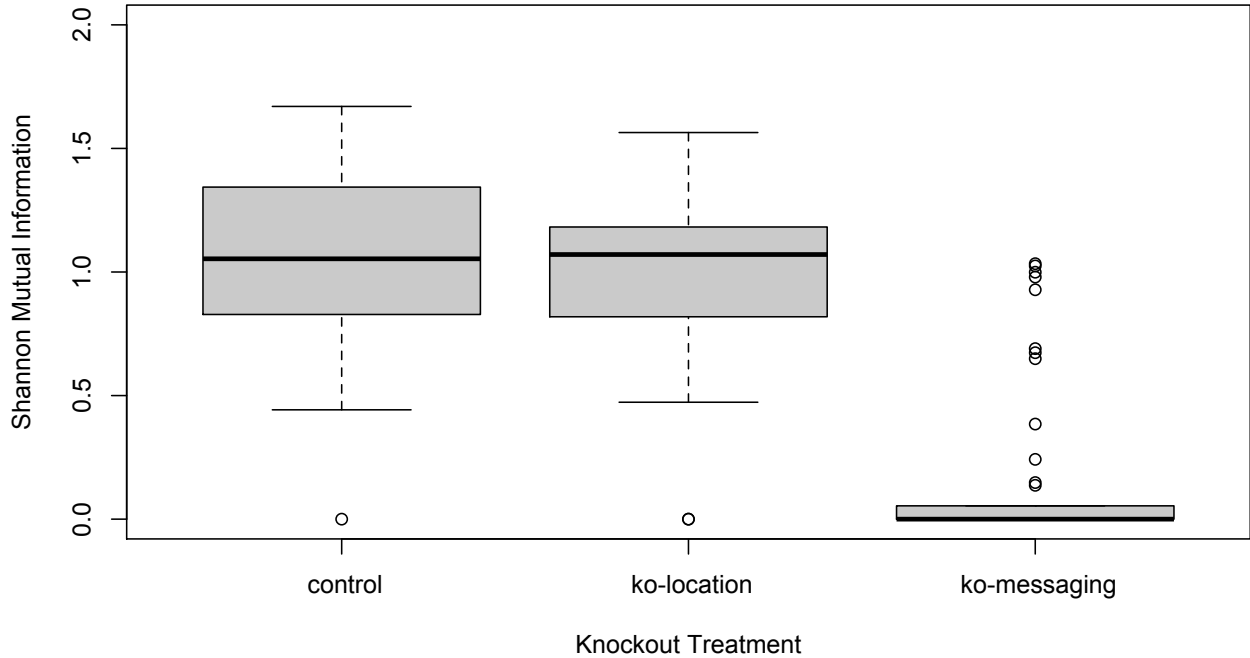


Figure 4.19: Boxes depict how the amount of Shannon mutual information varies as different knockout treatments are performed for a group that exhibits division of labor in the Logic-9 environment. Notably, knocking out location does not significantly affect the behavior of the majority of the groups. However, knocking out messaging has a tremendous effect and lowers the median amount of division of labor.

perform the operation $3 + 5 * 2 = 13$ and would then set its role-id to 17. This is a clever algorithm for making use of location information to perform division of labor.

4.2.4 Discussion

In this section, we described how we have used Avida, a digital evolution platform, to explore one hypothesis for the origin of division of labor. Specifically, we have found that the application of task-switching costs is sufficient to increase the amount of specialization and division of labor present within groups of organisms. We then performed additional experiments to assess the generality of these results and found that they were robust to environmental perturbations in terms of the criteria for successful groups and the type of tasks present within

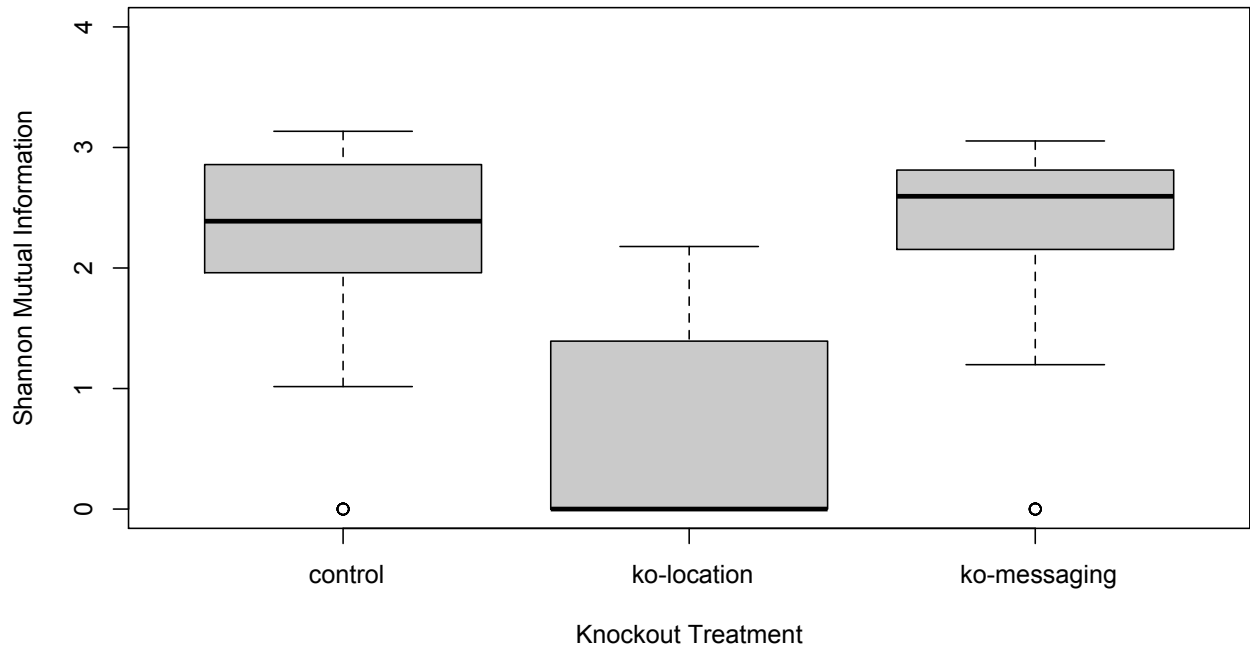


Figure 4.20: Boxes depict how the amount of Shannon mutual information varies as different knockout treatments are performed for a group that exhibits division of labor in the 25-role environment. Notably, knocking out messaging does not significantly affect the behavior of the majority of the groups. However, knocking out location has a tremendous effect and lowers the median amount of division of labor.

the environment. We also discovered support for the idea that efficiency gains through parallel processing may also favor the evolution of division of labor strategies. Lastly, we analyzed the behavior of the successful groups and determined they were coordinating their roles using message passing and location awareness depending on the environment. These results shed light on how task-switching costs present in the environment may give rise to division of labor strategies.

		column-ids (x)				
		0	1	2	3	4
row-ids (y)	0	0	1	2	3	4
	1	5	6	7	8	9
	2	10	11	12	13	14
	3	15	16	17	18	19
	4	20	21	22	23	24

Figure 4.21: The role-ids selected by the organisms in a successful group that performs 24 (out of 25 possible) different roles and exhibits division of labor in the 25-role environment. All but one id (0) are rewarded.

```

...
get-cell-xy      # loads the x and y coordinates into
                  # the AX and BX registers, respectively

add              # AX += BX
add              # AX += BX
add              # AX += BX
h-copy
add              # AX += BX
add              # AX += BX
set-role-id      # set role-id to the contents of the AX register
...

```

Figure 4.22: An excerpt of the genome used by a successful group evolved in the 25-role environment to perform 24 different roles. Coordination is achieved via a location-sensitive algorithm. Specifically, organisms set their role-ids to $x + 5 * y$

Chapter 5

Applications of Division of Labor: Problem Decomposition

Our previous studies shed light on the evolutionary conditions that give rise to division of labor strategies. In this chapter, we first describe a technique that we developed for problem decomposition [42] prior to our division of labor studies. However, in a fascinating twist, some of the groups that exhibit division of labor did so using an advanced form of problem decomposition that supersedes the abilities of our engineered effort. In the second part of this chapter, we describe the evolved strategies and the implications for research in evolutionary algorithms.

5.1 Engineered Problem Decomposition Approach

In this section, we describe an engineered technique for problem decomposition. Specifically, our technique evolves specialists that produce subcomponents and cooperate with other neighboring specialists to collect all of the subcomponents necessary to form a complete solution. Because an individual organism collects all of the subcomponents required for

the complete solution, this approach has the potential to be used to address problems that require more complex assembly procedures, which can also be evolved.

For this study, we provide instructions and infrastructure to allow organisms to cooperate in order to obtain subcomponents produced by neighboring organisms using *indirect reciprocity* [83]. Whereas *direct reciprocity* involves the exchange of altruistic donations during repeated interactions between the same two individuals, indirect reciprocity involves the exchange of donations for reputation, where future donations are directed toward individuals with a high reputation [83, 116]. In this study, an organism’s reputation is an accurate reflection of its behavior that is automatically updated. Specifically, an organism can donate a subcomponent to a neighboring organism. As a result of this donation, the donor organism’s reputation automatically increases. This improved reputation, in turn, increases the probability that the donor organism will subsequently receive donations of subcomponents from others. Thus, an organism can assemble a complete solution by combining the subcomponent(s) that it produced itself with those it has received from others.

We demonstrate the potential for this approach using a variant of the binary string cover problem presented in [94]. Whereas the original binary string cover problem considers a *set of individuals* to be a solution, we explicitly require an *individual* to cooperate with other neighboring individuals to assemble a complete solution. In this case, assembling a solution simply requires an organism to collect all of the subcomponents. First, to test whether this approach can evolve cooperation, we force individuals to be specialists that produce only one subcomponent and cooperate to solve the complete problem. Next we compare the performance of a population of generalists that each attempt to solve the entire problem to the performance of a population of specialists. On simpler problems, the population of generalists outcompete the population of specialists. However, as the complexity of the problem increases, the population of specialists outperforms the population of generalists. Lastly, we demonstrate that when allowed to be either generalists or specialists, the population

composition automatically changes to solve the problem most efficiently.

5.1.1 Methods

Here we provide an overview of the binary string production problem that we use to illustrate our technique, discuss the modifications to Avida that were necessary to allow specialists to cooperate, and provide the standard configurations used for our experiments.

The original binary string cover problem, which we are using a variation of, considers each individual in a population to represent a binary string x . The objective of each individual is to match as strongly as possible a set V of N binary vectors called *strings*, where each string v_i in V is of length l . Let x be a match value, where b_i is the number of bits exactly matched (value and placement) between x and v_i . Formally, x is defined as follows:

$$S(x, v_i) = \begin{cases} 0 & \text{if } b_i < \frac{l}{2} \\ (\frac{2b_i}{l} - 1)^2 & \text{if } b_i \geq \frac{l}{2} \end{cases} \quad (5.1)$$

Unless the strings in set V are identical, no single value of x will be able to perfectly optimize all objectives, and thus trade-offs must occur. The EA's solution to the binary string cover problem is a set of individuals in the final population [35, 94]. If individuals within the population specialize, then they are able to produce a better overall solution. Goings and Ofria have previously demonstrated the ability of Avida to develop specialists that cover multiple niches in this problem [35].

Whereas the original binary string cover problem statically composes subcomponents by considering a set of individuals (one from each species) to represent the solution, this variation increases the difficulty of the original problem by requiring each individual to produce or acquire all of the strings and assemble the complete solution. We refer to this variation as the *binary string production problem*. Thus, the objective of the individual is,

given a set V of binary strings, to produce multiple exact copies of V . We refer to a perfect copy of V as a *complete set*.

For our experiments, we use several variants of the binary string production problem, where each variant differs in complexity. Specifically, we define 4-bit, 8-bit, 16-bit, and 24-bit variants. Each variant has two strings that constitute V : the first string is all zeros and the second string is all ones. For example, the 4-bit variant uses strings 0000 and 1111 and the complete set is $\{0000, 1111\}$.

To enable organisms to specialize and cooperate to produce complete solutions, we extended the organism infrastructure, defined instructions to allow organisms to produce complete sets, and defined tasks that described the desired outputs. At a high level, an organism creates a string in its buffer and then produces copies of the string. The copies can be donated to increase its reputation or used as parts of a complete set. Tasks are associated with the quality of strings produced, quantity of strings produced, and also the quantity of complete sets produced.

Figure 5.1 (a) depicts the four key elements of an Avida organism that are used to produce solutions to the binary string production problem. First, an organism has a *buffer* in which it is able to construct strings. The buffer is initially empty and is exactly the length of a string. A pointer into the buffer determines where insertion will occur. The pointer starts at the first element, and moves one position each time a bit is inserted into the buffer. After the last element has been written, the pointer returns to the first element and future insertions overwrite existing bits. For example, Figure 5.1 (a) depicts a buffer of length 4 (for the 4-bit variant of the string production problem). Second, an organism has two counters that track the number of copies of each string that the organism has at its disposal, where a copy could be either produced by the organism or received as a donation. For example, Figure 5.1 (a) depicts the two counters (c_{0000} and c_{1111}). The number of copies of a string is limited to a user-defined amount. For these experiments, the limit is 10 copies per string. A copy can be

used as part of a complete set or as a donation. Third, an organism has a *tag* that identifies which vector it is best at producing. An organism initially inherits the tag of its parent. As the organism replicates, its tag and the tag of its offspring are updated to reflect the vector it is best at producing. Lastly, an organism has a reputation that is automatically updated as a result of the organism’s cooperative actions. An organism’s reputation is determined using a variant of a standing strategy [116] as follows ¹:

- **reputation = 0**: An organism has neither donated a string nor received a string. All organisms are born in this state.
- **reputation = -1**: An organism has not donated a string, but has received a string.
- **reputation = 1**: An organism has donated a string and may or may not have received a string.

Table 5.1 depicts the instructions we added to the standard Avida instruction set [85]. Instructions **insert-0** and **insert-1** insert a 0 or a 1 into the organism’s buffer. Figure 5.1 (b) depicts an organism after it executed **insert-0**. Instruction **prod-string** reads an organism’s buffer. If the string in the buffer perfectly matches one of the vectors in V , the organism’s counter for this vector is incremented by 1; otherwise, the instruction has no effect. For example, Figure 5.1 (c) depicts the buffer of an organism that has constructed string 0000 and then executed **prod-string**. The counter for the string 0000, c_{0000} , is incremented. The buffer remains the same after the **prod-string** instruction is executed. Thus, an organism may consecutively execute the instruction several times to create multiple copies of the same string.

The remainder of the new instructions allow organisms to cooperate through indirect reciprocity. To donate a string, an organism executes the instruction **donate-string**, which decrements the organism’s counter for the string and increments the corresponding counter

¹We also experimented with reputation strategies in which an organism’s reputation increased with each subsequent donation. These strategies did not qualitatively change our results.

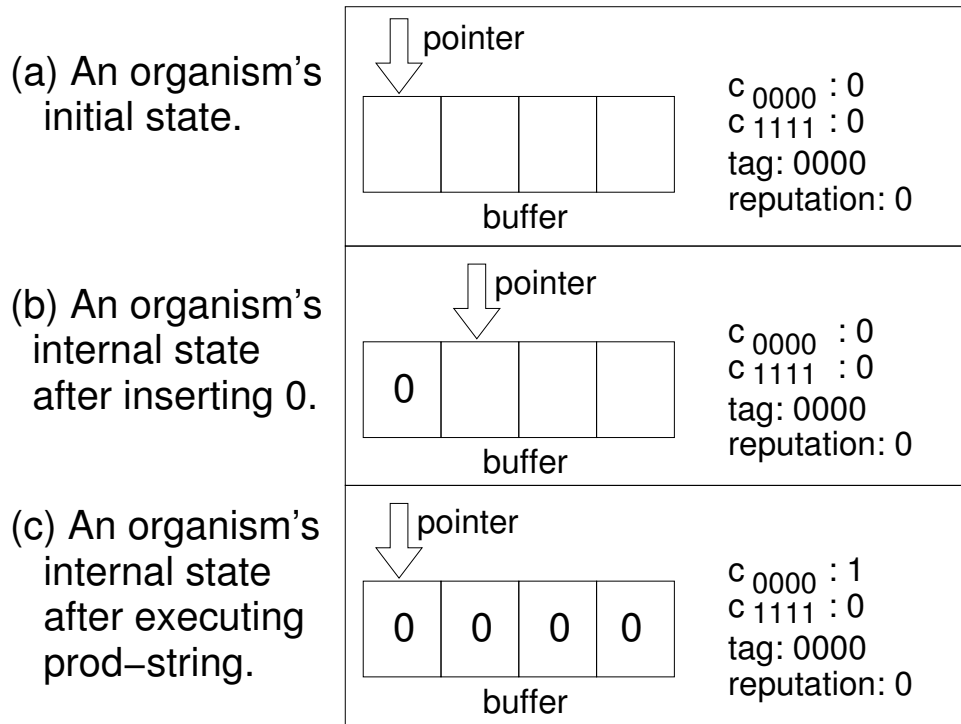


Figure 5.1: The infrastructure that allows Avida organisms to produce strings, donate strings, and cooperate using indirect reciprocity.

of the neighbor. An organism's tag determines which string it donates. For example, if an organism's tag is 0000, then the organism donates string 0000. Donations between organisms of the same tag would not assist organisms in constructing a complete set and thus are not allowed. Additionally, if an organism does not have any copies of the string it donates (e.g., $c_{0000} = 0$), then the donation does not occur and the counters and reputations of the organism and its neighbor remain unchanged.

Organisms are able to sense their own reputation and the reputation of their facing neighbor using the `get-reputation` and `get-neighbors-reputation` instructions, respectively. When these instructions are executed, the reputation value is placed in one of the organism's registers. Lastly, the `rotate-to-rep-tag` instruction rotates an organism to face the neighbor with the highest reputation of the opposite tag. If the organism does not have a neighbor of the opposite tag, then its facing remains unchanged. If multiple neighboring organisms of

equally high reputation all have the opposite tag, then the organism is rotated to face a random organism from this set.

Instruction	Description
insert-0	Insert 0 into the buffer
insert-1	Insert 1 into the buffer
prod-string	If the string in the buffer is a vector v_i , increment the counter of v_i
donate-string	Donate a copy of a string to the facing neighbor
get-reputation	Get the organism's reputation
get-neighbors-reputation	Get the neighbor's reputation
rotate-to-rep-tag	Rotate to the neighbor with a different tag and the highest reputation

Table 5.1: Additional Avida instructions that allow organisms to produce strings, donate strings, and selectively cooperate with neighbors on the basis of reputation.

We defined two types of tasks associated with resources that reward organisms for the binary string production problem. The first task, **MatchString**, is associated with a limited resource and is used to ensure that the population maintains the ability to produce both strings. The performance of an organism on **MatchString** is proportional to the quality of the string in its buffer when the organism replicates. A **MatchString** task is defined for both strings in V (e.g., **MatchString**₀₀₀₀ and **MatchString**₁₁₁₁) and the value of the task is defined using the same formula as the original bit string cover problem. Additionally, if the organism has produced the string that perfectly matches the vector itself, then it receives a score of 1. The formula is:

$$MatchString(org, v_i) = \begin{cases} 0 & \text{if } b_i < \frac{l}{2} \\ (\frac{2b_i}{l} - 1)^2 & \text{if } b_i \geq \frac{l}{2} \\ 1 & \text{if } c_{v_i} \geq 1 \end{cases} \quad (5.2)$$

The **MatchString** task also labels the organism with a tag that corresponds to the vector it is best at producing. Specifically, an organism’s tag corresponds to the vector whose **MatchString** task had the highest quality.

The second type of task, **CompleteSet**, is associated with an unlimited resource that rewards organisms for producing complete sets. Specifically, the task quality is equal to four times the number of complete sets plus the number of additional copies of either string. Task quality is partially defined based on strings that cannot be used as part of a complete set to reward organisms for production quantity. Formally, let c_{low} be the counter with the lowest balance, bal_{low} . Let c_{high} be the counter with the higher balance, bal_{high} . The formula for computing the reward of the **CompleteSet** function is:

$$CompleteSet(org) = (4\ bal_{low}) + (bal_{high} - bal_{low}) \quad (5.3)$$

Our experiments all used a common set of configurations. Each experiment comprised 20 replicate runs to account for the stochastic nature of the evolutionary process. Each run had 3,600 cells (and thus a maximum population size of 3,600 organisms) arranged in a toroidal topology. Each run started with two untagged asexual self-replicating organisms that constructed a string of the appropriate length for the variant of the problem being attempted. One organism constructed a random string and the other constructed the complement of the random string. Seeding the organisms with strings assists the organisms in solving the bit string production problem because, to receive the resources associated with completing a **MatchString** task, an organism must construct a string that has over 50% of the bits correct.

The starting length of the organisms was 100 instructions. When an organism replicated, each instruction had a 0.0075 probability of mutating as it was copied. Additionally, each genome had a 0.05 probability of an insertion mutation and a 0.05 probability of a deletion mutation. The offspring of an organism was placed in a random location in the torus. This

mass action replacement strategy ensures that neighboring organisms are unrelated and thus cooperation would not be simply because of kin selection [13, 34]. The runs lasted for 50,000 updates.

5.1.2 Results

In the remainder of this section, we demonstrate our problem decomposition technique on the bit string production problem. First, to ensure that organisms are able to dynamically cooperate to solve the problem, we require all organisms to be specialists and exploit the information produced by other organisms to cooperate. Because the specialist strategy incurs the overhead of cooperation, a generalist strategy may outperform the specialist strategy on simple problems. To test this hypothesis, we compare the performance of a population of specialists and a population of generalists on the bit string production problem. We then enable organisms to be generalists or specialists and demonstrate that Avida varies the population composition in response to the complexity of the problem.

Can specialists perform problem decomposition?

Our first set of experiments test whether Avida can evolve a population of specialists that cooperate to solve binary string production problems of varying complexity (i.e., 4-bit, 8-bit, 16-bit, and 24-bit strings). We force all organisms to be *strict specialists*, meaning that they produce strings that match at most one string and therefore must cooperate to create complete sets. To ensure all organisms are strict specialists, we modified the **prod-string** instruction so that an organism can produce copies only of the string with which it is tagged. Thus, if an organism is tagged with string 0000, then it can only produce copies of string 0000 and must receive string 1111 as a donation.

Although we provide the instructions necessary for the organisms to cooperate using indirect reciprocity, the population must evolve an algorithm that effectively uses these

instructions to produce complete sets. It is possible that a population could cooperate using a different technique. To isolate the critical information necessary to enable cooperation among unrelated specialists, we ran four treatments that varied the ability of the organisms to sense tags and reputations. Specifically, the four treatments are:

- Organisms do not have the ability to sense either tags or reputation. (no rep, no tags)
- Organisms have the ability to sense tags, but not the ability to sense reputation. (no rep, tags)
- Organisms have the ability to sense reputation, but not the ability to sense tags. (rep, no tags)
- Organisms have the ability to sense both reputation and tags. (rep, tags)

We ran 20 replicates for each variant of the binary string production problem and evaluated the results according to the number of complete sets produced and the number of organisms that produced complete sets. The relative ranking of the treatments was the same for all variants. Figure 5.2 depicts the results of varying the ability of the organisms to sense tags and reputations for the 16-bit variant. Each treatment produced significantly different results ($p < 0.05$, Mann-Whitney U test). Specifically, Figure 5.2 (a) depicts the number of organisms that produced complete sets and Figure 5.2 (b) depicts the number of complete sets produced by the organisms. The solid lines are the mean of 20 treatments and the dotted lines are one standard error of the mean. Overall, organisms without either tag or reputation information (green line with triangles) performed poorly; whereas, organisms with both tag and reputation information (blue line with circles) achieved both the largest number of organisms producing complete sets ($\sim 2,000$) and also the largest number of complete sets produced ($\sim 10,000$).

Counterintuitively, organisms with reputation information, but not tag information (black line with upside down triangles) produced fewer complete sets than organisms without any information. We performed further investigations and found out that, although these

organisms are producing fewer complete sets, their average fitness exceeds that of organisms without any information. Essentially, their fitness is greater because they are producing more copies of a single string. Additionally, organisms with tag information, but not reputation information (red line with squares) performed slightly better than organisms with no information, but not nearly as well as organisms with both reputation and tag information ($p < 0.05$, Mann-Whitney U test).

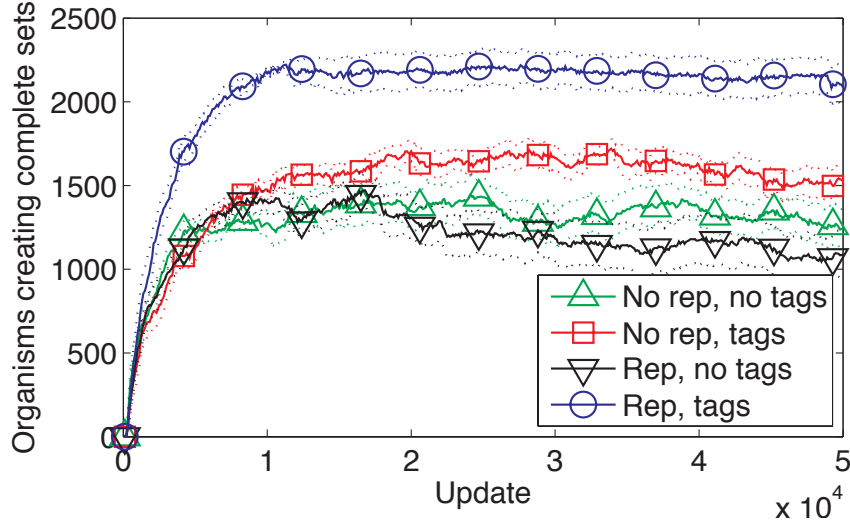
Based on these experiments, we conclude that the specialist strategy is affected by the organism’s ability to sense both tags and reputation. For the remainder of the experiments, organisms can sense both reputation and tag information.

How do specialists compare to generalists on performing problem decomposition?

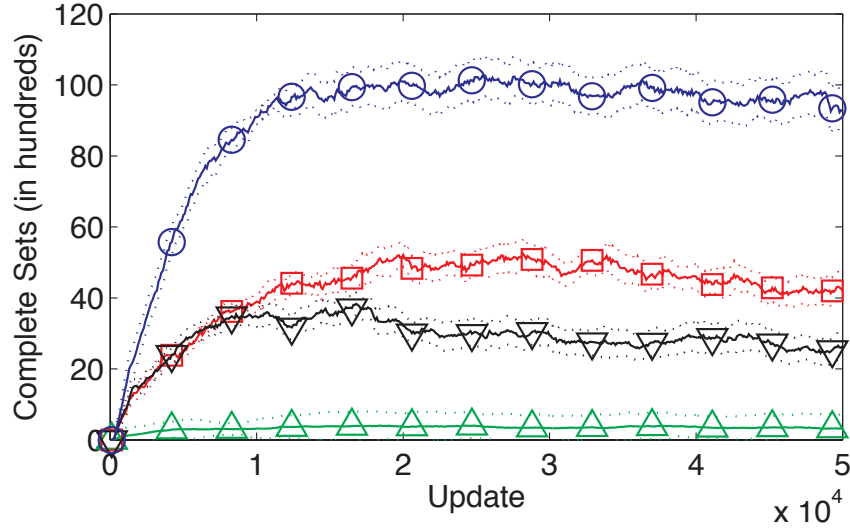
The primary motivation for using a problem decomposition approach is to address problems of increased complexity that are difficult for a standard EA to address. Next, we compare the performance of strict specialists to *strict generalists*, where strict generalists do not have a **donate-string** instruction and thus are unable to cooperate. To create a complete set, a strict generalist must first create one string (e.g., 0000) in its buffer, produce copies of the string, create the other string in its buffer (e.g., 1111), and then produce copies of it.

Our dual hypotheses for these experiments are that:

1. If a problem is simple, then a population of generalists will outperform a population of specialists, since it is easier to produce both strings than to cooperate. Thus, for simpler variants of the problem, the generalists treatments should both have more organisms that produce complete sets and have more complete sets produced than the specialist experiments.
2. If the problem is complex, then the specialists will outperform the generalists, since it is easier to cooperate than to produce both strings. Thus, the specialist experiments should both have more organisms that produce complete sets and have more complete



(a) Number of organisms that produce complete sets

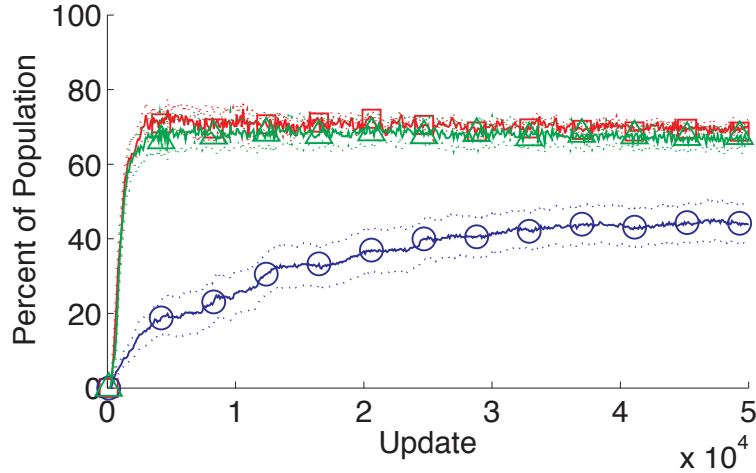


(b) Number of complete sets produced

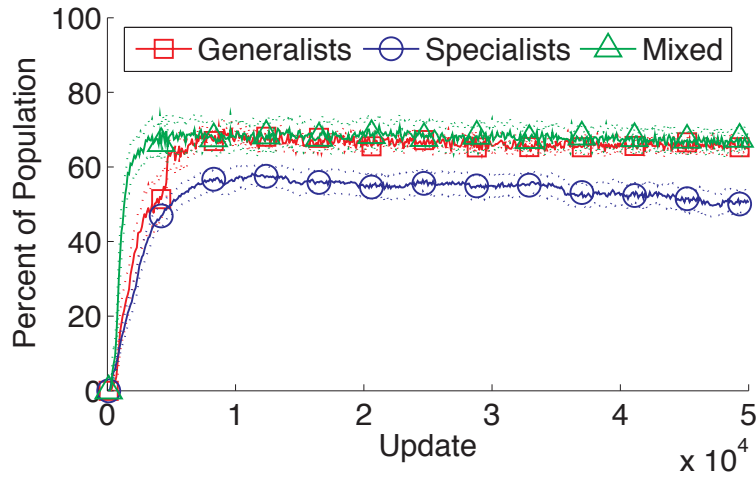
Figure 5.2: Number of organisms that produce complete sets (a) and the number of complete sets produced (b) with four different treatments that vary the tagging and reputation information available for the 16-bit variant. Populations where organisms have both sources of information significantly outperform populations that lack either.

sets produced than the generalist experiments for the more complex variants of the bit string production problem.

We tested these hypotheses by running 20 replicates of strict generalists and 20 replicates of strict specialists for each of the four variants of the bit string production problem. The



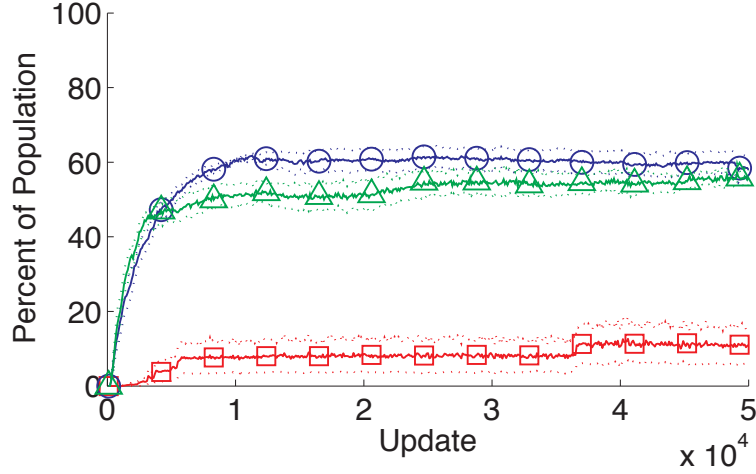
(a) 4-bit strings



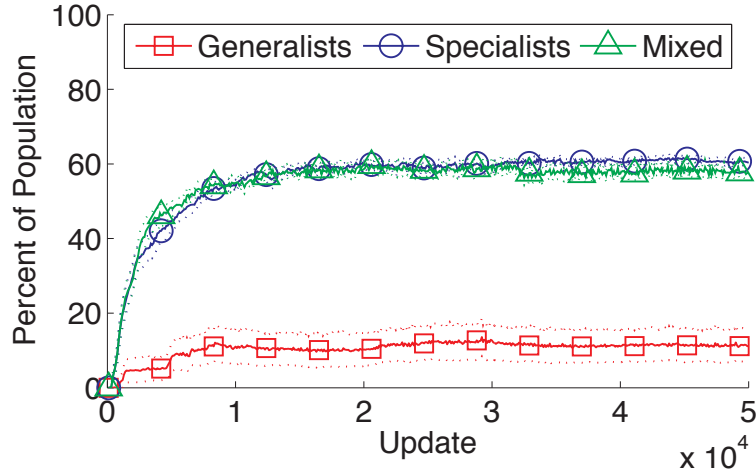
(b) 8-bit strings

Figure 5.3: The percentage of the population of organisms producing complete sets across the 4-bit and 8-bit string production problem variants. For simpler problems, generalists outperform specialists; whereas, for more complex problems, specialists outperform generalists. Mixed populations tend to perform at least as well as the isolated populations of generalists or specialists.

results of the experiments are depicted in Figures 5.3-5.6, where the red lines with squares represent the generalists and the blue lines with circles represent the specialists. For the simpler 4-bit and 8-bit variants, the generalists outperform the specialists both in terms of the number of organisms producing complete sets and also in terms of the number of complete sets produced ($p < 0.05$, Mann-Whitney U test). However, for the more complex 16-bit



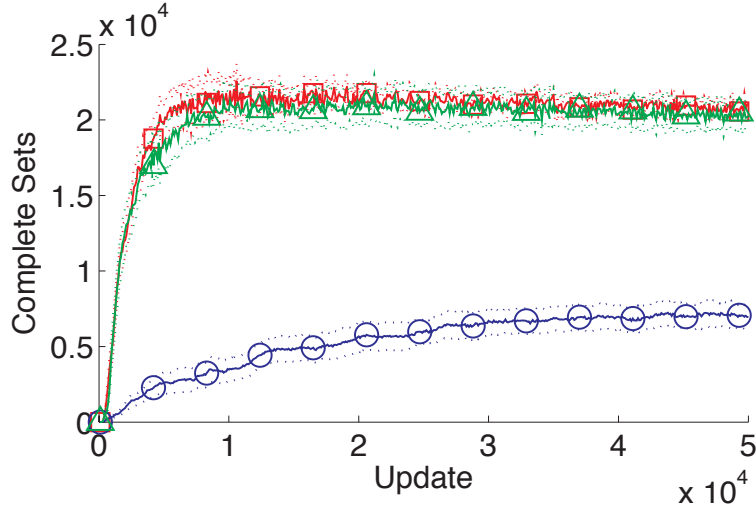
(a) 16-bit strings



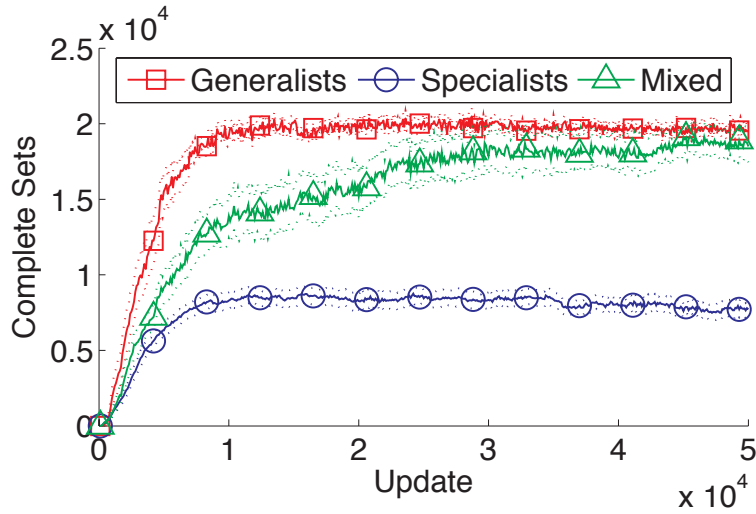
(b) 24-bit strings

Figure 5.4: The percentage of the population of organisms producing complete sets across the 16-bit and 24-bit production problem variants. For simpler problems, generalists outperform specialists; whereas, for more complex problems, specialists outperform generalists. Mixed populations tend to perform at least as well as the isolated populations of generalists or specialists.

and 24-bit variants, the specialists outperform the generalists ($p < 0.05$, Mann-Whitney U test). The performance of the specialists slightly increases with the complexity of the problem. Specifically, approximately the same number of specialist organisms (~ 2000) produced approximately the same number of complete sets ($\sim 10,000$) for the 8-bit, 16-bit, and 24-bit treatments. In contrast, the success of the generalists was dependent upon the complexity



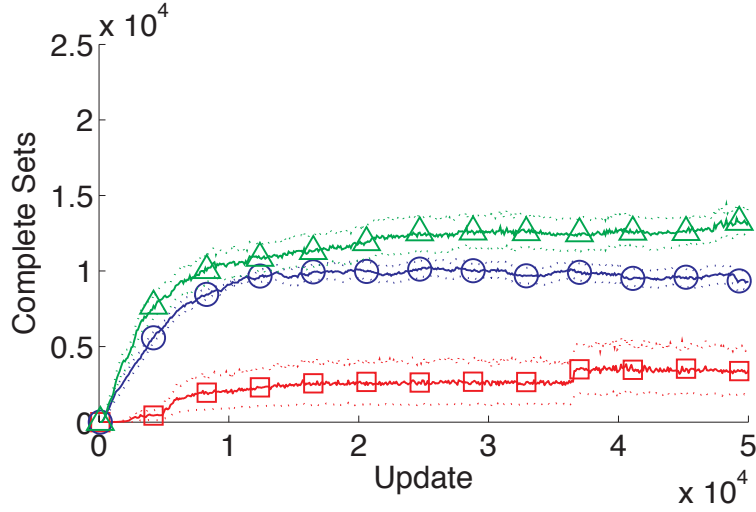
(a) 4-bit strings



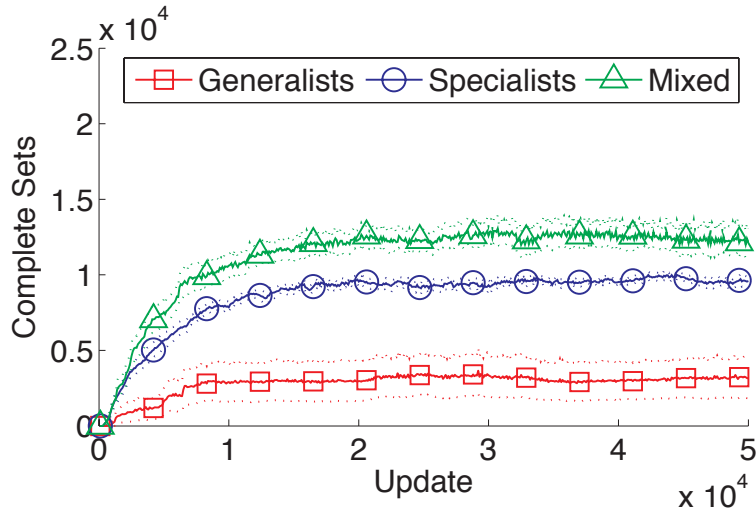
(b) 8-bit strings

Figure 5.5: The number of complete sets produced by generalists, specialists, and mixed populations of both generalists and specialists for the 4-bit and 8-bit string production problem variants. For simpler problems, generalists outperform specialists; whereas, for more complex problems, specialists outperform generalists. Mixed populations tend to perform at least as well as the isolated populations of generalists or specialists.

of the problem. For the 4-bit and 8-bit treatments, $\sim 2,400$ generalist organisms produced $\sim 20,000$ complete sets, but for the 16-bit and 24-bit treatments, ~ 400 generalists produced $\sim 4,000$ complete sets. These results support our hypotheses that complex problems favor specialists, whereas simpler problems favor generalists.



(a) 16-bit strings



(b) 24-bit strings

Figure 5.6: The number of complete sets produced by generalists, specialists, and mixed populations of both generalists and specialists for the 16-bit and 24-bit string production problem variants. For simpler problems, generalists outperform specialists; whereas, for more complex problems, specialists outperform generalists. Mixed populations tend to perform at least as well as the isolated populations of generalists or specialists.

When does specialization occur?

Ideally, a problem decomposition approach would evolve generalists or specialists depending on the complexity of the problem. To test the ability of our technique to automatically vary

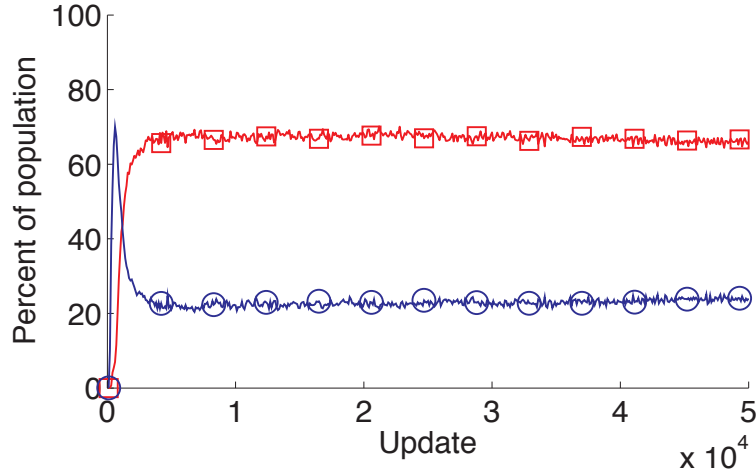
population composition, we enable both types of organisms to coexist within a population.

Our hypothesis for the next set of experiments is: Given a population in which organisms could be either generalists or specialists, the composition of the population will change depending on the complexity of the problem. Specifically, if the problem is simpler, then the generalists should dominate; if the problem is more complex, then the specialists should dominate. However, enabling organisms to be either generalists or specialists should not negatively affect the ability of the population to solve the problem.

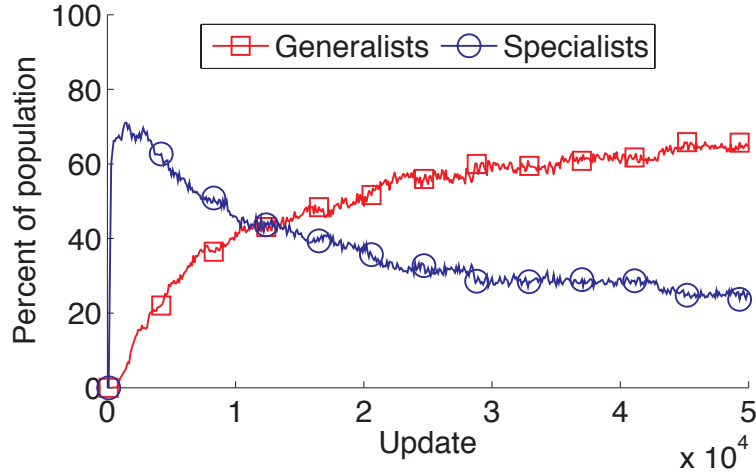
To test this hypothesis, we ran 20 replicates for each of the bit string production problem variants in which organisms could be either generalists or specialists. Specifically, we both enabled the organisms to donate and did not limit them to producing one string. For these results, we consider an organism to be a specialist if it produces only one string and to be a generalist if it produces both.

The green line with triangles in Figures 5.3-5.5 depict the results of a mixed population of generalists or specialists. In general, the performance of the mixed population toward the end of the run is equivalent to that of the isolated treatment (either generalist or specialist) that performs the best. Additionally, the number of complete sets produced by the mixed populations for the 16-bit and 24-bit variants is higher than that of the isolated populations of both specialists and generalists ($p < 0.05$, Mann-Whitney U test). Therefore, we conclude that the mixed population does not negatively affect Avida's ability to solve the problem.

The average population compositions from the four variants are depicted in Figures 5.7 and 5.8, where the red lines with squares represent the percent of organisms that are generalists and the blue lines with circles represent the percent of organisms that are specialists. As the complexity of the problem increases, so does the prevalence of specialists. Specifically, in the 4-bit and 8-bit variants, generalists tend to dominate the population ($p < 0.05$, Mann-Whitney U test). However, in the 16-bit and 24-bit variants, the specialists tend to dominate the population ($p < 0.05$, Mann-Whitney U test).



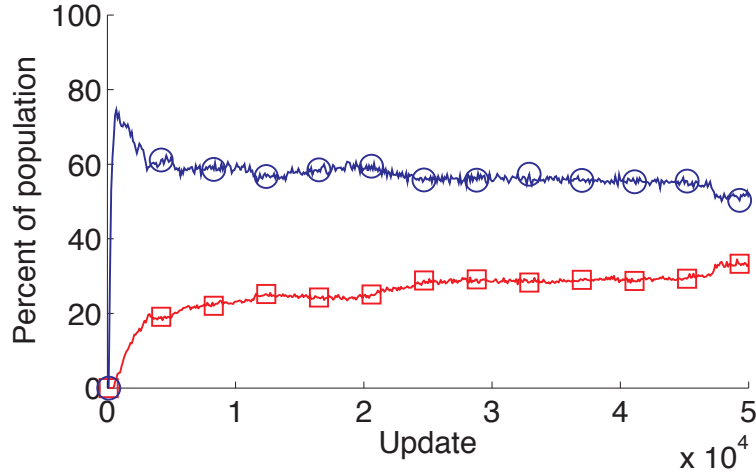
(a) 4-bit strings



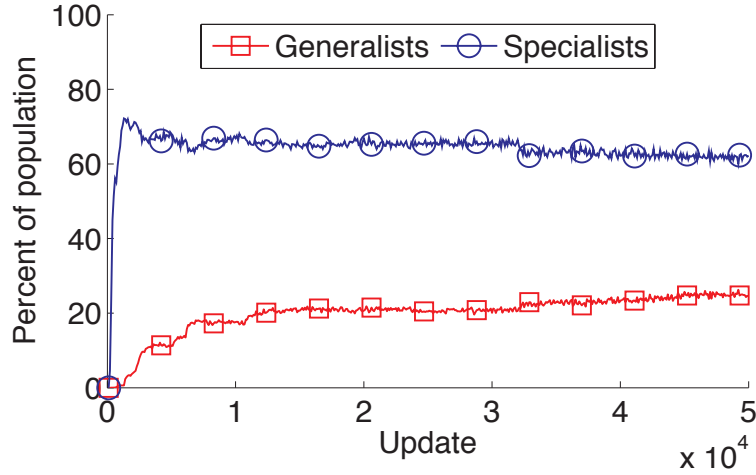
(b) 8-bit strings

Figure 5.7: The population composition of mixed populations comprising both generalists and specialists for the 4-bit and 8-bit variants of the bit string production problem. In general, the more simple variants have a predominantly generalist population and the more complex variants have a predominantly specialist population.

Although the average population compositions make it appear as if the number of generalists in a population gradually increases, for the individual replicates, once a generalist strategy for producing complete sets is found, it quickly sweeps the population. Thus, in many cases, a specialist strategy tended to thrive early on, probably because it is simpler to evolve, but was quickly replaced when a generalist strategy evolved. Three possible factors



(a) 16-bit strings



(b) 24-bit strings

Figure 5.8: The population composition of mixed populations comprising both generalists and specialists for the 16-bit and 24-bit variants of the bit string production problem. In general, the more simple variants have a predominantly generalist population and the more complex variants have a predominantly specialist population.

that could influence the ability of generalist strategy to sweep the population are: First, because generalists do not rely on others for donations of subcomponents, they are a more robust strategy. Second, generalist organisms are, most likely, stingy in the sense that they do not donate. Figure 5.9 depicts the percent of *altruists* (organisms that donate a string) in the population, the percent of specialists, and the percent of generalists for the 16-bit vari-

ant with a mixed population. The percent of altruists almost perfectly tracks the percent of specialists, but appears unrelated to the percent of generalists. Because generalists do not donate, as the number of generalists increases, it becomes increasingly challenging for the specialists to find neighboring organisms with which to cooperate. Third, generalists likely benefit from some donations from specialists, since a generalist is indistinguishable from a specialist that has not yet donated in that both have a reputation of 0.

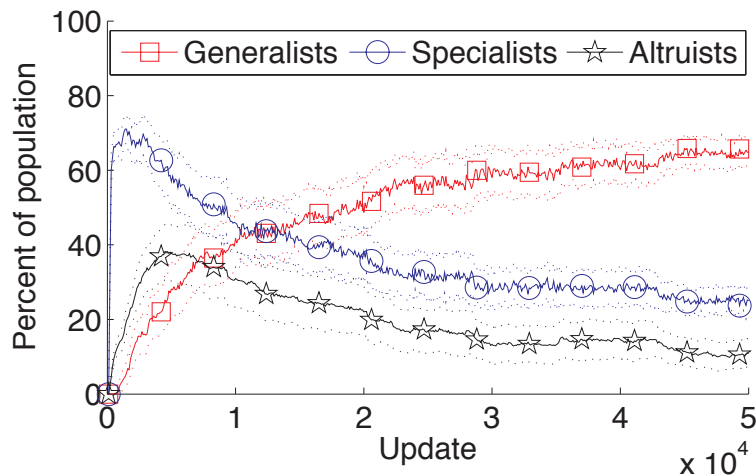


Figure 5.9: The percent of organisms that are generalists, specialists, and altruists. The percent of altruists in the population closely tracks the number of specialists in the population, indicating that generalists are not making donations.

5.1.3 Conclusions

In this section, we have presented a digital-evolution based approach to problem decomposition. Our approach enables individuals to be either generalists that independently solve the problem or specialists that each contribute a portion of the overall solution. We have demonstrated the feasibility of this technique using several variants of the binary string production problem. As the complexity of the problem increases, the population composition automatically changes from predominantly generalists to predominately specialists. However, once a generalist strategy evolves within the population, it quickly dominates. Some limitations

of this approach are that it relies upon human guidance to decompose the problem and to specify how the solution should be formed from the problem components.

5.2 Evolved Problem Decomposition Approach

While studying how task-switching costs influence the evolution of division of labor, we analyzed two groups that exhibited high degrees of division of labor. Both groups exhibited problem decomposition strategies, even though problem decomposition was not directly rewarded by either of our experiments. Here, we present our analyses of the groups and then describe the implications for evolutionary algorithms.

5.2.1 Group case study 1.

As part of our preliminary study for the role of task-switching costs in the evolution of division of labor, we uncovered an example of emergent problem decomposition. This preliminary study [44] differs from later work presented as a part of this dissertation in Section 4.2 in that the organisms used a simpler form of self replication, the limited resources did not have an outflow, there were only five possible logic tasks that the organisms could perform, and the resource requirement for group replication was set to 350 resources. The group that we analyzed resulted from the 50-cost treatment. This group performed 4 (out of 5 possible) different types of tasks.

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 group. 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 group with the knockout version of the genotype and assessed its

```

...
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 5.10: A snippet of the genome of a organism that when placed with clones of itself in a group performs division of labor and problem decomposition.

performance. The group's performance was not affected by the removal of location information, age of the organism, nor age of the group. However, if the group's messaging capability was disabled, the group 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 group's strategy.

Second, we isolated how messaging was being used by the group. Figure 5.10 provides 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: Organisms perform the more complex tasks when they receive messages containing the solutions to the simpler tasks. These organisms were cooperatively solving the problem. Organisms performed higher level tasks using the solutions computed by organisms for the lower-level tasks.

5.2.2 Group case study 2.

Our second case study resulted from an analysis of a group evolved as part of the task-switching experiments described in Section 4.2. The environment in which this group evolved differed from the previous case study in that organisms could perform up to nine logic tasks, limited resources were configured with an outflow, and the resource requirement for group replication was set to 500 resources. Additionally, in our earlier experiment, organisms replicated by executing a `repro` instruction. In this experiment, organisms were required to include the instructions to copy each instruction of their genome to an offspring organism, which is a much more complex process. The group that we analyzed resulted from the 50-cost treatment. This group performed 7 (out of 9 possible) different types of tasks in the control analyses, but was not able to replicate when its messaging capabilities were knocked out.

We focused on isolating how messaging was being used by the group. To start, we examined what messages organisms were sending. Messages consist of two numbers designated by the organisms. Initially, when we equipped the organisms with message sending capabilities, we envisioned them using the messages as semaphores to coordinate which organism was performing which task. However, in this case, the organisms instead sent messages containing either the solution to a task, or a number slightly off from the result of the task (i.e., a result obtained by adding or subtracting a number less than three). Specifically, the messages sent by the organisms contained the solution to all seven of the tasks performed

by the organisms.

A preliminary study of the genotype for this group indicated that it was significantly more challenging to decipher than the genotype used for the original case study. As a result, we needed to proceed more methodically to figure out how it functioned. There are several possible ways that the organisms could use messages containing task results: (1) organisms did not use the contents of the messages, but rather just the presence of them to coordinate behavior, (2) organisms received a result and then performed the task, (3) organisms used the result as part of a calculation for a more complex task – the logic tasks are building blocks for one another. To assess whether organisms were using the contents of the messages, we randomized the numbers upon which they performed logic tasks, which had the effect of making task results organism specific. This modification ensured that messages could still be sent and received, but that contents were unable to contribute to the task performance of receiving organisms. With this modification, the group was unable to perform any tasks, thus ruling out the first possibility. To distinguish between the second and third options we examined whether organisms performed a task they received the result of; they did not. We studied the third possibility in greater detail by tracing the behavior of every group member, including their messaging interactions. We discovered that organisms sent task results, which were then used as building blocks by the receiving organisms for computing more advanced tasks. For example, in the early part of the group life when there are only two organisms, the offspring organism sends the parent organism the results of performing task **not**. Using this result, the parent organism performs some additional calculations to perform tasks **ornot** and **andnot**, sending out the results. Thus, organisms are using the task results they receive in messages as building blocks for computing the solutions to more complex tasks.

While our study focused on assessing whether division of labor arose more frequently in the presence of higher task-switching costs, the behavior of the group indicates a far more complex strategy than we had anticipated. In this instance, group members perform problem

decomposition by breaking complex logic tasks into their constituent simpler tasks, computing the results for simpler tasks, sharing the task results, and assembling the solutions to produce the results for more complex tasks. This strategy is far more complex and coordinated than a simple approach in which different group members specialize on performing a single task in isolation. Intriguingly, this strategy also reflects how some natural organisms perform division of labor. For example, leafcutter ants (*Atta vollenweideri*) decompose the task of tending to fungi into majors that cut leaves, mediae that move leaves from the tree to the colony, and minors that tend to the fungal gardens [53]. Both the strategy evolved by the colony in this study and the strategy evolved by the leafcutter ants exhibit problem decomposition and assembly line processing of task material. This result suggests that the efficiency advantages afforded by problem decomposition may favor the evolution of division of labor.

5.3 Implications

While problem decomposition has been speculated to enable EAs to address more challenging problems than any one individual can solve, it is a challenging behavior to evolve. Ideally, we would like for EAs to automatically decompose the problem, solve the subproblems, and assemble the sub-solutions to form the overall solution. Our engineered approach relied upon developer guidance to: (1) specify how the problem should be decomposed (i.e., which strings should be created), (2) how organisms could cooperate to exchange information (i.e., donating strings), and (3) how the problem was to be reassembled (i.e., a complete set was formed by having both strings). In contrast, the groups evolved as part of our task-switching studies automatically: (1) decomposed the problem with minimal guidance (i.e., while we did specify all the logic functions, the organisms figured out that complex tasks could be broken down into simpler tasks), (2) solved the subproblems (i.e., performed the math necessary

for the lower-order logic functions), and (3) stitched together the overall solution (i.e., they used messaging to send the results of less complex tasks and evolved algorithms that made use of these messages for assembling more complex tasks).

Notably, these results were achieved even though we were not directly rewarding for division of labor or problem decomposition – we simply allowed for it and these mechanisms were the most effective ways organisms found to solve the group task. Another result from these studies is that when there was a more direct method to solve the problem, such as was the case for the 25-role studies, the organisms did not evolve to use problem decomposition. Additionally, these results underscore the importance of providing organisms with flexible mechanisms, such as messaging, that enable communication. One unintended strength of the messaging system used for these studies was that it did not prescribe what the contents of the messages should be, nor how organisms should process the incoming messages. These case studies suggest that imposing evolutionary conditions that facilitate division of labor may also result in individuals finding the means to exhibit problem decomposition when it is an effective method for solving the overall challenge.

Chapter 6

Conclusions

Division of labor, where individuals specialize on specific roles and cooperate to survive, is a hallmark strategy of a diverse set of groups within nature, including bacteria, slime molds, multicellular organisms, eusocial insects, eusocial mammals, and human economies [8, 16, 21, 32, 53, 55, 81, 98, 108, 111, 131]. As such an intriguing biological phenomena, scientists have labored to explain the mechanisms used to implement division of labor, as well as the reasons for its success. While researchers are able to experiment with modern groups that exhibit division of labor in order to understand the modern adaptive benefits of such a strategy, exploring the evolutionary conditions that give rise to division of labor remains a challenge because of the long generation times and lack of historical data for natural systems. In this dissertation, we have used Avida to explore evolutionary hypotheses for the division of labor. Additionally, we have demonstrated that these studies may enable the development of evolutionary algorithms that can address more complex problems.

6.1 Contributions

In summary, there are four main contributions of my research:

1. In preliminary studies that we conducted using a game theoretic approach [46] and Avida [43, 44, 45] we found evidence that evolutionary computation techniques can be used to evolve groups of organisms that exhibit division of labor. Our studies with genetically heterogeneous groups revealed that multilevel selection is a sufficient pressure to result in division of labor, even when the individual rewards for tasks are greatly disparate. Our studies with clonal groups revealed that the mechanisms used to achieve division of labor were environmentally dependent, but reflect strategies commonly observed in nature. This work provided the foundation for our studies of the evolutionary conditions under which division of labor is favored.
2. Next, we addressed a controversial hypothesis within biology: the temporal polyethism pattern exhibited by a wide range of eusocial insects can result from the risks associated with aging and certain types of tasks. We explored this question in the context of two and three tasks with different risks associated with them. In these studies, individuals performed the more risky task later in their lifetime, thus lending support to the hypothesis.
3. Subsequently, we examined the role of task-switching costs in the evolution of division of labor. The reduction of task-switching costs as a result of increased specialization has been proposed as a potential efficiency benefit provided by division of labor strategies. While this hypothesis is difficult to test in natural systems, since task-switching costs may include the time it takes to travel between tasks, morphological changes that must be made, and psychological changes that occur, we were able to address it more explicitly using digital evolution. We created an environment in which the cost of switching tasks was configurable. We found support for the hypothesis that as task-switching costs increased, groups of organisms exhibited increasingly greater amounts of division of labor. Additionally, this study resulted in support for another division of

labor hypothesis: that parallel processing may be an advantage provided by division of labor [41, 44].

4. Lastly, in analyzing the groups that exhibited division of labor as the result of the application of task-switching costs, we discovered that groups were employing a problem decomposition strategy. Problem decomposition strategies have been proposed as an approach to enabling evolutionary algorithms to address more complex challenges. Thus, our studies shed light on how to apply evolutionary pressures to produce groups that exhibit problem decomposition to address group-level tasks [41, 44]. This approach may provide the basis for a new type of genetic algorithm that is explicitly targeted at more complex problems.

6.2 Future Work

There are many possible future directions for division of labor research. In this section, I briefly discuss several intriguing directions for future work that include exploring how division of labor insights can be applied to evolutionary algorithms and expanding the scope of evolutionary questions that we explore.

6.2.1 Problem Decomposition in Embodied Agents

As part of my post-graduation plans, I intend to apply the pressures we used to evolve a group of organisms that perform problem decomposition to other problems including the design of robot-control algorithms.

For example, we are interested in applying our division of labor insights to evolve robot controllers that exhibit problem decomposition. The first project we plan to address is evolving groups of robots that must specialize and coordinate their behavior to collect two different types of resources. Specifically, the robots will be placed into an environment in

which there are two light beacons, where one beacon emits a green light and the other emits a red light. These beacons represent different resources present in the environment that can be collected by the robots if they flash their own corresponding red or green light within a certain radius surrounding the beacon. The objective of the group of robots is to collect 25 red resources and 25 green resources as quickly as possible. To confirm the results of our task-switching work, namely that increased task-switching costs promote division of labor strategies, we can run three different treatments. In the first treatment, the red and green beacons will be positioned close together enabling the robots to be generalists that may emit alternating flashes. In the second and third treatments, the beacons will be positioned a medium and significant distance apart, respectively. Our hypothesis is that as the beacons move further apart the robots will increasingly evolve division of labor strategies.

For this study, we can leverage an existing framework for developing controllers that makes use of the NEAT [115] and HyperNEAT [114] algorithms for evolving neural networks that control robots. In previous work, we used simulated robot controllers evolved using NEAT and HyperNEAT to manipulate groups of robots that spread out to cover an area [62]. During the evolutionary process, the solutions will be evaluated in simulation. We will then load the best evolved solutions into e-puck robots for testing. This project will extend that work by focusing on division of labor strategies and by including the step of loading robot controllers onto physical robots for additional testing. Future work will use these robots to perform complex tasks that require problem decomposition.

6.2.2 Evolution of Division of Labor

Building upon our current results, we can also test several other aspects of the evolution of division of labor. The first aspect we plan to investigate is the effect of the behavior of the ancestor organism. Currently, our experiments begin with a genome that gives rise to a specialist organism in that it only performs one task. Mutations to the genome during group

replication give rise to organisms that perform multiple tasks and in some cases exhibit a division of labor strategy where the organisms coordinate the roles they perform. However, there is good reason to believe that the ancestor organism of eusocial colonies was a generalist that performed all tasks necessary for survival, rather than a specialist [53]. To test this aspect of the evolution of division of labor, we could evolve several generalist organisms that perform all nine logic function in the standard Avida environment and then use them as the ancestor organisms for our runs. Because historical contingency regarding the genetic structure of the ancestor organism is likely to play a role in the evolution of division of labor, we would need to start with several different evolved generalist organisms. It is unlikely that using a generalist ancestor would invalidate our results, since this modification simplifies the problem – to exhibit division of labor, the organisms would need to evolve only coordination capabilities, rather than task capabilities and coordination capabilities.

One area of investigation in division of labor studies is the role of genetic variability [53, 109]. While genetic variability can be seen as a source of potential conflict within the colonies for the reproductive role, it also may be used as a method for selecting tasks [109]. There is evidence that certain genetic lines within a colony are more prone to perform a specific task than others [109]. Within our studies exploring the pressures that give rise to division of labor strategies, we have focused on clonal colonies and thus have eliminated all sources of genetic variability. However, we have the ability to experiment with the amount of genetic variability present within a colony and observe its effects. We would expect that large amounts of genetic variability could be harmful to a group’s ability to cooperate on a group task, although, in small amounts, genetic variability may enhance division of labor strategies.

Broadening the scope of our investigations, a third area we intend to investigate is one of the other potential benefits of division of labor: increased efficiency via learning [20, 108]. Our current approach has examined the role of task-switching costs. However, we could

adopt a similar approach to studying the role of learning. Specifically, the simplest approach would be to apply a large penalty the first time an organism performs a new type of task. This penalty would be reduced each time the organism performs the task, implementing efficiency gains through learning. A more advanced approach could include increasing the penalty if the organism has not performed the task recently, which would be analogous to forgetting infrequently performed actions.

6.2.3 Major Transitions in Evolution

While we have focused on the role division of labor has played as a strategy within biological groups, division of labor also has a principal role in the major transitions in evolution [111]. These major transitions in evolution occur when formerly distinct individuals join together in a higher-level unit that functions as a single reproductive entity [33, 60, 99, 100, 111]. These transitions can be *fraternal* (Figure 6.1 (a)¹), where genetically similar individuals (i.e., close kin) create a super-organism and then differentiate to perform various tasks, or *egalitarian* (Figure 6.1(b)) in which formerly distinct organisms create a super-organism that replicates all of its genetic material [95, 96]. For example, fraternal transitions include single cells transitioning into multicellular organisms and distinct insects transitioning into eusocial colonies; egalitarian transitions include the “eukaryotic alliance between a host cell and its mitochondria” [95]. These transitions raise evolutionary questions regarding why formerly distinct individuals would choose to cooperate with others and, once they did, how this arrangement persisted.

While biologists have discovered evidence of these transitions, there is still a great deal of mystery regarding the evolutionary forces that favor these transitions. Why would individuals give up their ability to reproduce independently? One hypothesis regarding an explanation for fraternal transitions is that transitioning between reproduction and other

¹Figure inspired by [60].

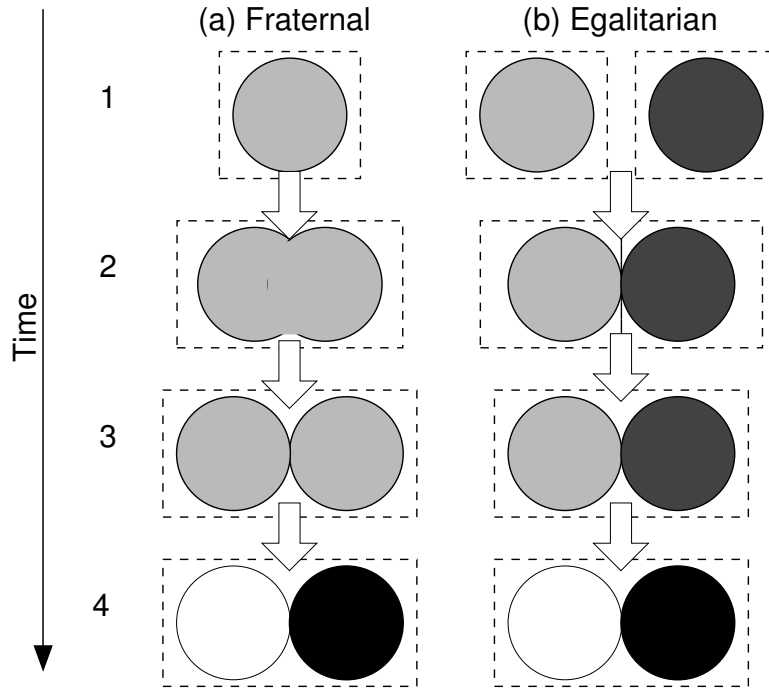


Figure 6.1: Illustration of fraternal and egalitarian transitions. Circles represent individuals, boxes represent groups, arrows represent steps. For a fraternal transition (a), an organism (1), produces two daughter organisms (2), these daughters choose to remain in a group (3), and eventually differentiate their behavior (4). For an egalitarian transition (b), two organisms with different phenotypes (1) are proximally near one another (2) and choose to form a group (3). Finally, further evolution differentiates the behavior of the organisms within the group (4).

tasks may be time consuming, if not impossible. For example, single cells of *Volvox* algae use their flagella to explore their medium and absorb resources; however, they are not able to reproduce while flagellated. Thus, some species have undergone a fraternal transition to form groups in which some flagellated individuals move the group, while other non-flagellated individuals reproduce [81]. To study this topic, we can expand on our task-switching work to make the cost of transitioning between other tasks and reproducing variable. Our hypothesis is that as this cost rises, groups will increasingly evolve to exhibit reproductive division of labor. One potential outcome of this study is support for Queller's hypothesis that reproductive division of labor can result from conditionally expressed instructions that are dependent on location or the behavior of nearby individuals [96].

Egalitarian transitions in evolution occur when genetically distinct individuals cease to replicate separately and instead begin to replicate as a group, where the genetic material of both formerly independent individuals is copied together to the new group [95]. One fascinating example of an egalitarian transition occurred in a colony of amoeba that when infected with a harmful bacteria eventually evolved to not only neutralize the effect of the bacteria, but also to depend upon the bacteria for survival [57]. These transitions give rise to the question of under which conditions would formerly independent individuals choose to form a group that replicates as a single entity [60]? Examples in nature suggest that one or more of the participants in an egalitarian transition become more efficient as a result of a close physical association with another individual and, in order to strengthen the fidelity of their bond, evolve to replicate as a group. To explore questions surrounding egalitarian transitions, we can relax two of the constraints imposed by my other division of labor studies. First, whereas my other experiments forcibly place organisms into groups, in these experiments organisms may choose to form or dissolve bonds with other organisms that result in groups. Second, whereas my other experiments automate part of the group life cycle, these experiments seek to understand how organisms may evolve their own group life cycle.

APPENDICES

Appendix A

Overview of Additional Research Projects

Over the course of my Ph.D. studies, I have had the opportunity to work on a variety of interesting projects that are not a part of, but rather inspired my dissertation research. I started as a software engineer, began using evolutionary computation to address challenges within software engineering, and then became fascinated with evolutionary computation itself. In this appendix, I provide an overview of some of my other research projects and contributions.

Software engineers are faced with developing and detecting errors in high assurance systems. The stakes of such development efforts are high, not only because of the substantial costs of error correction, but also because failures may lead to injury, and, in the worst case, loss of life [77]. To address some of the challenges present in developing these systems we turned to evolutionary computation approaches.

A.1 Avida-MDE

Increasingly, high-assurance applications rely on autonomic systems to react and respond to changing environmental concerns; examples include critical infrastructure protection and transportation systems. As such, it is essential that the corresponding autonomic reactions do not put the system into an inconsistent state or deliver unacceptable behavior. In an effort to promote separation of concerns, we consider an *autonomic system* to comprise a collection of (non-adaptive) *target systems* and a set of adaptations that realize transitions among target systems in response to environmental changes. A key challenge with developing autonomic systems is to identify robust and resilient target systems that handle the various, sometimes adverse, environmental conditions [59]. A second challenge is to develop more abstract representations of these systems as a means to manage the complex functional and adaptation requirements and the corresponding implementations [59]. Model-driven development (MDD) technology [104] supports the systematic transformation of such abstract representations (graphical models) into more detailed models or formal specifications, and eventual generation of the corresponding code. While MDD offers an attractive development approach, the efforts required to create or revise models in order to start the MDD process (i.e., initial requirements-level or early design models) can be error prone and difficult to automate. To address both challenges, we proposed a biologically-inspired approach to automatically generate requirements-level behavioral models for target systems, where the models support user-defined scenarios and satisfy formally specified safety-critical properties [37, 38, 40, 79].

Several approaches have been developed to model target systems, including: architecture description languages [2, 10], goal models [27, 70], and state machines [70, 135]. Additionally, significant progress has been achieved in synthesizing behavioral models from scenarios [3, 50, 69, 122, 124, 128] and from formally specified properties [58, 93, 122]. Scenarios and

properties form the lower and upper bounds on the possible behavior of the system [122], respectively; that is, a scenario depicts what at least one path of behavior through a model must satisfy, and properties indicate what all paths through a model must satisfy. However, there are many behavioral models that exist in the solution space between these boundaries. One drawback of existing approaches is that they limit the exploration of the solution space for target systems to those envisioned by the developer who hand-crafted the model or designed the synthesis algorithm. Since autonomic systems are reacting to a large number of environmental and internal conditions, whose combined effect may not be fully understood at development time, we need target systems that can deliver the essential services in a more robust and resilient fashion than human developers might envision [59].

In this work, we described an approach to generating behavioral models for target systems that leverages a technique used by living organisms to adapt to changing environmental conditions: *evolution*. Our approach is based on digital evolution [87]. Whereas genetic algorithms and genetic programs evaluate each individual in the population and explicitly select individuals to move to the next generation, the evolution of digital organisms is more open-ended and thus more likely to discover novel and previously unknown solutions. Until recently, digital evolution has been used primarily by biologists to address questions that are difficult or impossible to study with organic life forms [72, 86]. However, digital evolution also provides a means to *harness* the power of evolution and apply it to problems in science and engineering [36, 64, 79, 86], sometimes discovering strikingly clever and innovative solutions to complex problems.

Our approach uses digital evolution to generate behavioral models of target systems. Each digital organism is treated as a generator of UML state diagrams that depict system behavior: when the organism executes, it constructs an in-memory representation of a *behavioral model* that comprises one or more state diagrams. An organism’s ability to compete for resources is directly related to whether the generated behavioral model meets the criteria specified

by the developer. Mutations introduced during replication produce organisms with varying abilities to compete, while competition for resources gives rise to a population of organisms that produce increasingly better solutions. To implement this method, we extended the Avida digital evolution platform [87] in three key ways. First, we enabled each organism to have *instinctual knowledge*, namely, information embedded in the organism at birth. In this study, the instinctual knowledge comprises information about the UML class diagram elements (that describe the structural elements) and an optional set of *seed state diagrams* (that is, UML state diagrams specifying the behavior of the existing system). Second, we enhanced the Avida instruction set to include instructions that enable an organism to use its instinctual knowledge to construct transitions in one or more state diagrams. Third, we enabled Avida to use third-party software to assess the behavioral models generated by organisms. For example, we make use of the Spin model checker [54] to check the Avida-generated state diagrams for adherence to safety-critical properties.

The primary contribution of this work is a digital evolution technique for automatically generating behavioral models for an autonomic system. In contrast to other manual and synthesis techniques, this technique generates an array of possible solutions that are unbiased by human preconceptions. We illustrated this technique on the evolution of target systems for an autonomic robot navigation system [61].

A.2 Marple

Model-driven engineering, which successively refines models from analysis to design and then automatically generates code [104], can accidentally propagate errors in the analysis model constructed early in the development process to code. Formally analyzing a model for adherence to its requirements can detect errors in the *sanctioned behavior*, the known, acceptable, and possibly required behavior of the model. This analysis, however, does not

detect errors in *latent behavior*, the unspecified and potentially unwanted behavior of the model; these errors could then be propagated to the implementation and even deployed.

One approach to ensuring that models used for model-driven development provide the desired behavior is to analyze them for adherence to system requirements [76, 80, 118]. This analysis, however, does not detect errors in *latent behavior*, the unspecified and potentially unwanted behavior of the model; these errors could then be propagated to the implementation and even deployed. Uchitel et. al have proposed an approach for detecting one form of latent behavior called implied scenarios as part of the process of synthesizing a model from scenarios [123]. However, preexisting UML models cannot make use of this technique. Three broad categories of approaches have been developed to produce properties that could be used for analysis: *Requirements discovery approaches* (e.g., [78]) examine testing and deployment artifacts to detect missing or erroneous properties; process improvements have been proposed as part of these approaches. *Refinement-based approaches* (e.g., [73]) infer properties from formally specified goals or requirements. Lastly, *specification generation techniques* (e.g., [1, 11, 12, 26, 28, 49, 56, 127, 133]) infer properties from a representation of a system (e.g., a model or code) or a derivative of the system (e.g., execution traces). Several previously developed specification generation approaches are able to infer temporal logic properties from a model [11, 49], code [127], or execution traces [12, 133]. For these approaches, the developer identifies a part of the system behavior to explore, either by restricting the exploration to a portion of the code [127], or by explicitly selecting the states, events, and variables that are of interest [11, 12, 49, 133]. One ramification of having the developer guide the exploration is that the unexplored portions of the system may still conceal latent unwanted behavior. Ideally, developers would like to maximize both automation of property discovery and coverage, while minimizing the number of properties that must be examined.

To address this challenge, we proposed an evolutionary-computation approach called

MARPLE¹ to automatically generate properties that specify the latent behavior of UML models comprising an instance (class) diagram and multiple state diagrams [39]. Evolutionary computation methods, such as genetic algorithms and genetic programming, have achieved considerable success, in some cases producing human-competitive designs [68]. Each evolutionary algorithm experiment comprises a population of individuals. Over the course of many generations, where each generation is subjected to natural selection, mutation, and crossover, the evolutionary algorithm seeks to optimize according to a *fitness function* that describes one or more objectives. For this approach, we use a recently developed technique called *novelty search*, where the objective is not to find one optimal solution, but rather to find a suite of sufficiently different solutions [71]. We use novelty search to enable MARPLE to produce properties that maximize coverage of the model’s behavior, while minimizing human effort.

MARPLE uses novelty search to discover a set of properties that describe a UML model, where these properties describe behavior not explicitly stated in the requirements and may, in fact, be unacceptable latent behavior. Specifically, each individual within MARPLE represents a property created by instantiating one of the five most commonly occurring specification patterns [22] in the form of Linear Temporal Logic (LTL). Instantiating a pattern involves replacing the placeholders with evolved boolean propositions, where a proposition is created using attribute and operation information from a UML instance diagram of the system. Because the propositions can include conjunctives and disjunctives, the set of possible propositions is unlimited and too large for brute force search methods to explore. During the evolutionary process, mutations and crossover produce different LTL properties that may be satisfied by the UML model. The novelty of a property is assessed using the Spin model checker [54]. Specifically, the state space of the shortest *witness trace* (i.e., path that sup-

¹MARPLE is named after Miss Marple, Agatha Christie’s detective who was famous for detecting latent human behavior.

ports the property) through the Spin representation of the model is compared to the state spaces of other properties.² If a *novel region* of the model state space is discovered (i.e., a region of the state space that has not been explored by previously evaluated properties), then the property is assigned a higher fitness value and MARPLE searches the new region more thoroughly. However, if a property explores a previously explored region of the state space, then it is assigned a low fitness value and MARPLE does not search the region as thoroughly. In this way, MARPLE discovers properties that cumulatively describe the behavior of the model. For readability purposes, the properties generated by MARPLE are presented to the developer in natural language [67] for assessment. The generated properties can be used by the developer to refine the requirements specifications (to explicitly sanction the latent behavior) or to modify the UML model (to remove unwanted latent behavior).

Overall, our approach enables developers to automatically explore UML models for properties representing potentially unwanted latent behavior. We illustrated our approach by using MARPLE to discover the unwanted latent behavior of models for an automobile door locking system obtained from one of our industrial collaborators. To further validate our approach, we have also applied it to a robot navigation system [61] and sought feedback from our industrial collaborators. Industrial collaborators, including researchers at Ford Motor company, provided feedback that MARPLE was a much needed means for detecting unwanted behavior in models of high assurance systems.

A.3 Conclusions

The challenges we faced in addressing software engineering challenges led to my interest in the development and improvement of evolutionary computation techniques in general. These challenges motivated my interest in understanding how division of labor insights could be

²Spin provides configuration options for generating the shortest path, which is how we produce the shortest witness trace.

applied to evolutionary computation techniques and in creating new problem decomposition approaches.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Mithun Acharya, Tao Xie, Jian Pei, and Jun Xu. Mining API patterns as partial orders from source code: from usage scenarios to specifications. In *ESEC-FSE '07*, pages 25–34, New York, NY, USA, 2007. ACM.
- [2] Robert Allen, Rémi Douence, and David Garlan. Specifying and analyzing dynamic software architectures. *Lecture Notes in Computer Science*, 1382, 1998.
- [3] Rajeev Alur, Kousha Etessami, and Mihalis Yannakakis. Inference of message sequence charts. *IEEE Trans. Softw. Eng.*, 29(7):623–633, 2003.
- [4] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.
- [5] Benjamin E. Beckmann, Philip K. McKinley, David B. Knoester, and Charles Ofria. Evolution of cooperative information gathering in self-replicating digital organisms. In *Proceedings of the First IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, pages 65–76, 2007.
- [6] Eric Bonabeau, Guy Theraulaz, and Jean-Louis Deneubourg. Fixed response thresholds and the regulation of division of labor in insect societies. *Bulletin of Mathematical Biology*, 60:753–807, 1998.
- [7] Josh C. Bongard. The legion system: A novel approach to evolving heterogeneity for collective problem solving. In *Proceedings of the European Conference on Genetic Programming*, pages 16–28, London, UK, 2000. Springer-Verlag.
- [8] John Tyler Bonner and Kenneth 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.
- [9] Andrew F. G. Bourke and Nigel R. Franks. *Social Evolution in Ants*. Princeton University Press, 1995.
- [10] Carlos Canal, Ernesto Pimentel, and José M. Troya. Specification and refinement of dynamic software architectures. In *Proceedings of the TC2 First Working IFIP Conference on Software Architecture (WICSA1)*, WICSA1, pages 107–126, Deventer, The Netherlands, The Netherlands, 1999. Kluwer, B.V.
- [11] William Chan. Temporal-logic queries. In *CAV '00: Proceedings of the 12th International Conference on Computer Aided Verification*, pages 450–463, London, UK, 2000. Springer-Verlag.

- [12] Richard M. Chang, George S. Avrunin, and Lori A. Clarke. Property inference from program executions. Technical Report UM-CS-2006-26, University of Massachusetts, 2006.
- [13] Jeff Clune, Heather J. Goldsby, Charles Ofria, and Robert T. Pennock. Selective pressures for accurate altruism targeting: Evidence from digital evolution for difficult-to-test aspects of inclusive fitness theory. *Royal Society*, 2010.
- [14] David Cornforth and Michael Kirley. Cooperative problem solving using an agent-based market. In *Genetic and evolutionary computation conference (GECCO)*, pages 60–71, Seattle, Washington, 2004.
- [15] Margaret J. Couvillon and Anna Dornhaus. Small worker bumble bees (*Bombus impatiens*) are hardier against starvation than their larger sisters. *Insectes Sociaux*, pages 1–5, 2010.
- [16] Bernard J. Crespi. The evolution of social behavior in microorganisms. *Trends in Ecology & Evolution*, 16(4):178–183, April 2001.
- [17] Robert Dahl. *On Political Equality*. Yale University Press, New Haven, CT, USA, 2006.
- [18] Daniel C. Dennett. The new replicators. In Mark Pagel, editor, *The Encyclopedia of Evolution*, volume 1, pages E83–E92. Oxford University Press, 2002.
- [19] Francis D. DeWaal. *Chimpanzee Politics: Power and Sex Among Apes*. Johns Hopkins University Press, May 2000.
- [20] Anna Dornhaus. Specialization does not predict individual efficiency in an ant. *PLoS Biol*, 6(11):e285, 11 2008.
- [21] J. Emmett Duffy. The ecology and evolution of eusociality in sponge-dwelling shrimp. In *Genes, Behaviors and Evolution of Social Insects*. Hokkaido University Press, 2003.
- [22] Matthew B. Dwyer, George S. Avrunin, and James C. Corbett. Patterns in property specifications for finite-state verification. In *Proceedings of the 21st International Conference on Software Engineering*, pages 411–420. IEEE Computer Society Press, 1999.
- [23] Mary Jane West Eberhard. The evolution of social behavior by kin selection. *The Quarterly Review of Biology*, 50(1):1–33, March 1975.
- [24] E.M. Eddy. The germ line and development. *Developmental Genetics*, 19:287–289, 1996.
- [25] Émile Durkheim. *The Division of Labor in Society*. Free Press, New York, NY, USA, 1893.

- [26] Michael D. Ernst, Jake Cockrell, William G. Griswold, and David Notkin. Dynamically discovering likely program invariants to support program evolution. *IEEE Transactions on Software Engineering*, 27(2):99–123, 2001.
- [27] Martin S. Feather, Stephen Fickas, Axel Van Lamsweerde, and Christophe Ponsard. Reconciling system requirements and runtime behavior. In *IWSSD '98: Proceedings of the 9th International Workshop on Software Specification and Design*, 1998.
- [28] Cormac Flanagan and K. Rustan M. Leino. Houdini, an annotation assistant for ESC/Java. In *FME '01: Proceedings of the International Symposium of Formal Methods Europe on Formal Methods for Increasing Software Productivity*, pages 500–517, London, UK, 2001. Springer-Verlag.
- [29] Dario Floreano, Sara Mitri, Stphane Magnenat, and Laurent Keller. Evolutionary Conditions for the Emergence of Communication in Robots. *Current Biology*, 17:514–519, 2007.
- [30] Kevin R. Foster, Tom Wenseleers, Francis L.W. Ratnieks, and David C. Queller. There is nothing wrong with inclusive fitness. *Trends in Ecology and Evolution*, 21:599–600, 2006.
- [31] N. R. Franks, C. Tofts, and A. B. Sendova-Franks. Studies of the division of labour: neither physics nor stamp collecting. *Animal Behaviour*, 53(1):219 – 224, 1997.
- [32] Herbert Gintis, Samuel Bowles, Robert Boyd, and Ernst Fehr. *Moral Sentiments and Material Interests: On the Foundations of Cooperation in Economic Life*. MIT Press, Cambridge, MA, USA, 2005.
- [33] Peter Godfrey-Smith and Benjamin Kerr. Gestalt-switching and the evolutionary transitions. *The British Journal for the Philosophy of Science*, 2011. Accepted.
- [34] Sherri Goings, Jeff Clune, Charles Ofria, and Robert T. Pennock. Kin-selection: The rise and fall of kin-cheaters. In *Ninth International Conference on Artificial Life*, pages 303–308, Boston, MA, 2004.
- [35] Sherri Goings and Charles Ofria. Ecological approaches to diversity maintenance in evolutionary algorithms. In *IEEE Symposium on Artificial Life*, Nashville, TN, 2009.
- [36] H. J. Goldsby, D. B. Knoester, B.H.C. Cheng, P. K. McKinley, and C. A. Ofria. Digitally evolving models for dynamically adaptive systems. In *Proceedings of the ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, Minneapolis, Minnesota, May 2007.
- [37] Heather J. Goldsby and Betty H. C. Cheng. Automatically generating behavioral models of adaptive systems to address uncertainty. In *Proceedings of ACM/IEEE 11th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, Toulouse, France, September 2008.

- [38] Heather J. Goldsby and Betty H. C. Cheng. Avida-MDE: A digital evolution approach to generating models of adaptive system behavior. In *proceedings of Genetic and Evolutionary Computation Conference (GECCO 2008)*, Atlanta, Georgia, July 2008.
- [39] Heather J. Goldsby and Betty H. C. Cheng. Automatically discovering properties that specify the latent behavior of uml models. In *Proceedings of ACM/IEEE 13th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, Oslo, Norway, September 2010.
- [40] Heather J. Goldsby, Betty H. C. Cheng, Philip K. McKinley, David B. Knoester, and Charles A. Ofria. Digital evolution of behavioral models for autonomic systems. In *Proceedings of the 5th International Conference on Autonomic Computing (ICAC 2008)*, Chicago, Illinois, June 2008.
- [41] Heather J. Goldsby, Anna Dornhaus, and Charles Ofria. The evolution of division of labor in digital organisms. In *preparation*, 2011.
- [42] Heather J. Goldsby, Sherri Goings, Jeff Clune, and Charles Ofria. Problem decomposition using indirect reciprocity in evolved populations. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, Atlanta, Georgia, July 2009.
- [43] Heather J. Goldsby, David B. Knoester, Jeff Clune, Philip K. McKinley, and Charles Ofria. The evolution of division of labor. In *Proceedings of the European Conference on Artificial Life (ECAL)*, Budapest, Hungary, September 2009.
- [44] Heather J. Goldsby, David B. Knoester, and Charles Ofria. Evolution of division of labor in genetically homogeneous groups. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, Portland, Oregon, July 2010.
- [45] Heather J. Goldsby, David B. Knoester, and Charles Ofria. Specialization in heterogeneous groups. In *preparation*, 2010.
- [46] Heather J. Goldsby, Karthik Panchanathan, and C. David Navarrete. Brave leaders and opportunistic followers: A division of labor model. In *preparation*, 2010.
- [47] Faustino J. Gomez and Risto Miikkulainen. Solving non-markovian control tasks with neuroevolution. In *In Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 1356–1361, 1999.
- [48] Root Gorelick and Susan M. Bertram. Quantifying division of labor: borrowing tools from sociology, sociobiology, information theory, landscape ecology, and biogeography. *Insectes Sociaux*, 54(2):105–112, 2007.
- [49] Arie Gurfinkel, Marsha Chechik, and Benet Devereux. Temporal logic query checking: A tool for model exploration. *IEEE Transactions on Software Engineering*, 29(10):898–914, 2003.

- [50] David Harel, Hillel Kugler, and Amir Pnueli. Synthesis revisited: Generating statechart models from scenario-based requirements. In *Formal Methods in Software and Systems Modeling*, 2005.
- [51] Friedrich A. Hayek. *Individualism and Economic Order*. Routledge and Kegan Paul, London, UK, 1948.
- [52] Fatemeh Heidary, Mohammad Reza Vaeze Mahdavi, Farshad Momeni, Bagher Minaii, Mehrdad Rogani, Nader Fallah, Roghayeh Heidary, and Reza Gharebaghi. Food inequality negatively impacts cardiac health in rabbits. *PLoS ONE*, 3(11):e3705, 2008.
- [53] Bert Hölldobler and Edward O. Wilson. *The superorganism: the beauty, elegance, and strangeness of insect societies*. WW Norton & Company, 2009.
- [54] Gerard J. Holzmann. *The Spin Model Checker, Primer and Reference Manual*. Addison-Wesley, 2004.
- [55] Jennifer M. Jandt and Anna Dorhaus. Spatial organization and division of labour in the bumblebee *bombus impatiens*. *Animal Behavior*, 77:641–651, 2009.
- [56] Ralph Jeffords and Constance Heitmeyer. Automatic generation of state invariants from requirements specifications. *SIGSOFT Softw. Eng. Notes*, 23(6):56–69, 1998.
- [57] Kwang W. Jeon. Development of cellular dependence on infective organisms: Microcurricular studies in amoebas. *Science*, 9(4039):1122–1123, 1972.
- [58] Barbara Jobstmann and Roderick Bloem. Optimizations for LTL synthesis. In *FMCAD '06: Proceedings of the Formal Methods in Computer Aided Design*, 2006.
- [59] Jeffrey O. Kephart and David M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.
- [60] Benjamin Kerr and Joshua Nahum. Setting the stage for an egalitarian major transition: The evolution of restraint. 2011. In Press.
- [61] Minseong Kim, Suntae Kim, Sooyong Park, Mun-Taek Choi, Munsang Kim, and Hassan Gomaa. UML-based service robot software development: a case study. In *ICSE '06: Proceeding of the 28th international conference on Software engineering*, pages 534–543, 2006.
- [62] David B. Knoester, Heather J. Goldsby, and Philip K. McKinley. Neuroevolution of mobile ad hoc networks. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, Portland, Oregon, July 2010.
- [63] David B. Knoester, Philip K. McKinley, Benjamin Beckmann, and Charles Ofria. Directed evolution of communication and cooperation in digital organisms. In *Proceedings of the European Conference on Artificial Life (ECAL)*, Lisbon, Portugal, September 2007.

- [64] David B. Knoester, Philip K. McKinley, and Charles Ofria. 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.
- [65] David B. Knoester, Philip K. McKinley, and Charles Ofria. Cooperative network construction using digital germlines. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2008.
- [66] David B. Knoester, Andres J. Ramirez, Philip K. McKinley, and Betty H. C. Cheng. Evolution of robust data distribution among digital organisms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2009.
- [67] Sascha Konrad and Betty H. C. Cheng. Real-time specification patterns. In *Proceedings of the International Conference on Software Engineering (ICSE05)*, St Louis, MO, USA, May 2005.
- [68] John R. Koza, Martin A. Keane, Matthew J. Streeter, William Myrdlowec, Jessen Yu, and Guido Lanza. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Springer, 2003.
- [69] Ingolf Kruger, Radu Grosu, Peter Scholz, and Manfred Broy. From MSCs to state-charts. In *DIPES'98*. Kluwer., 1999.
- [70] Alexei Lapouchnian, Sotirios Liaskos, John Mylopoulos, and Yijun Yu. Towards requirements-driven autonomic systems design. In *DEAS '05: Proceedings of the 2005 Workshop on Design and Evolution of Autonomic Application Software*, pages 1–7, St. Louis, MO, USA, 2005.
- [71] Joel Lehman and Kenneth Stanley. Exploiting open-endedness to solve problems through the search for novelty. In S. Bullock, J. Noble, R. Watson, and M. A. Bedau, editors, *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 329–336. MIT Press, Cambridge, MA, 2008.
- [72] Richard E. Lenski, Charles Ofria, Robert T. Pennock, and Christoph Adami. The evolutionary origin of complex features. *Nature*, 423:139–144, 2003.
- [73] Emmanuel Letier. *Reasoning about Agents in Goal-Oriented Requirements Engineering*. PhD thesis, Louvain-la-Neuve, Belgium, 2001.
- [74] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10, 1966.
- [75] Peter Lichodziejewski and Malcolm 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.

- [76] Johan Lilius and Ivan Porres Paltor. vUML: A tool for verifying UML models. In *Proceedings of the 14th IEEE International Conference on Automated Software Engineering*, page 255, Washington, DC, USA, 1999. IEEE Computer Society.
- [77] Robyn R. Lutz. Targeting safety-related errors during software requirements analysis. In *SIGSOFT'93 Symposium on the Foundations of Software Engineering*, 1993.
- [78] Robyn R. Lutz and Inés Carmen Mikulski. Requirements discovery during the testing of safety-critical software. In *ICSE '03: Proceedings of the 25th International Conference on Software Engineering*, 2003.
- [79] Philip McKinley, Betty H.C. Cheng, Charles Ofria, David Knoester, Benjamin Beckmann, and Heather Goldsby. Harnessing digital evolution. *IEEE Computer*, 41(1):54–63, 2008.
- [80] William E. McUmbler and Betty H. C. Cheng. A general framework for formalizing UML with formal languages. In *Proceedings of the IEEE International Conference on Software Engineering (ICSE01)*, Toronto, Canada, May 2001.
- [81] Richard E. Michod. The group covariance effect and fitness trade-offs during evolutionary transitions in individuality. *Proceedings of the National Academy of Sciences*, 103(24):9113–9117, 2006.
- [82] Linda Northrop, Peter Feiler, Richard P. Gabriel, John Goodenough, Rick Linger, Tom Longstaff, Rick Kazman, Mark Klein, Douglas Schmidt, Kevin Sullivan, and Kurt Wallnau. Ultra-Large-Scale Systems - The Software Challenge of the Future. Technical report, Software Engineering Institute, Carnegie Mellon, June 2006.
- [83] Martin A. Nowak. Five rules for the evolution of cooperation. *Science*, 314(5805):1560–1563, December 2006.
- [84] Charles Ofria and Christoph Adami. Evolution of genetic organization in digital organisms. In *Proc. of DIMACS workshop Evolution as Computation*, pages 167–175, Princeton, NJ, 1999.
- [85] Charles Ofria, Christoph Adami, and Travis C. Collier. Design of evolvable computer languages. *IEEE Transactions in Evolutionary Computation*, 6:420–424, 2002.
- [86] Charles Ofria, Christoph Adami, and Travis C. Collier. Selective pressures on genomes in molecular evolution. *J. Theor. Biology*, 222:477–483, 2003.
- [87] Charles Ofria and Claus O. Wilke. Avida: A software platform for research in computational evolutionary biology. *Journal of Artificial Life*, 10:191–229, 2004.
- [88] George F. Oster and Edward O. Wilson. *Caste and ecology in the social insects*. Princeton Univ Pr, 1979.

- [89] Richard E. Page Jr. and Joachim Erber. Levels of behavioral organization and the evolution of division of labor. *Naturwissenschaften*, 89(3):91–106, 2002.
- [90] Andres Perez-Urbe, Dario Floreano, and Laurent Keller. Effects of group composition and levels of selection in the evolution of cooperation in artificial ants. In *European Conference on Artificial Life (ECAL)*, pages 128–137, 2003.
- [91] Torsten Persson and Guido Tabellini. Is Inequality Harmful for Growth? Theory and Evidence. *American Economic Review*, 1994.
- [92] Nathan Pike and William A. Foster. *Ecology of Social Evolution*. Springer Berlin Heidelberg, 2008.
- [93] Nir Piterman, Amir Pnueli, and Yaniv Sa’ar. Synthesis of reactive(1) designs. In *In Conference on Verification, Model Checking, and Abstract Interpretation*, pages 364–380, 2006.
- [94] Mitchell A. Potter and Kenneth A. DeJong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evol. Comput.*, 8(1):1–29, 2000.
- [95] David C. Queller. Cooperators since life began. *Quarterly Review of Biology*, pages 184–188, 1997.
- [96] David C. Queller. Relatedness and the fraternal major transitions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 355(1403):1647, 2000.
- [97] David C. Queller and Joan E. Strassmann. Kin selection and social insects. *BioScience*, 48(3):165–175, March 1998.
- [98] David C. Queller and Joan E. Strassmann. Eusociality. *Current Biology*, 13(22):R861–863, 2003.
- [99] Paul Rainey and Benjamin Kerr. Cheats as first propagules: a new hypothesis for the evolution of individuality during the transition from single cells to multicellularity. *Bioessays*, 32:872–880, 2010.
- [100] Paul Rainey and Benjamin Kerr. Conflicts among levels of selection as fuel for the evolution of individuality. 2011. In Press.
- [101] Gene Robinson. Regulation of division of labor in insect societies. *Annual Review of Entomology*, 37:637–665, 1992.
- [102] Simon K. Robson and Samuel N. Beshers. Division of labour and ‘foraging for work’: simulating reality versus the reality of simulations. *Animal Behaviour*, 53(1):214–218, 1997.
- [103] Paul Schmid-Hempel. Worker castes and adaptive demography. *Journal of Evolutionary Biology*, 5(1):1–12, 1992.

- [104] Douglas C. Schmidt. Model-Driven Engineering. *IEEE Computer*, 39(2), 2006.
- [105] Thomas D. Seeley. Honey bee colonies are group-level adaptive units. *The American Naturalist*, 150:S22–S41, July 1997.
- [106] Thomas D. Seely. Adaptive significance of the age polyethism schedule in honeybee colonies. *Behavioral Ecology and Sociobiology*, 11(4):287–293, 1982.
- [107] Ana Sendova-Franks and Nigel R. Franks. Task allocation in ant colonies within variable environments (a study of temporal polyethism: experimental). *Bulletin of Mathematical Biology*, 55(1):75–96, 1993.
- [108] Adam Smith and Alan B. Krueger. *The Wealth of Nations*. 1776.
- [109] Chris R. Smith, Amy L. Toth, Andrew V. Suarez, and Gene E. Robinson. Genetic and genomic analyses of the division of labour in insect societies. *Nature Reviews Genetics*, 9(10):735–748, 2008.
- [110] John Maynard Smith. Byte-sized evolution. *Nature*, 355(6363):772–773, 1992.
- [111] John Maynard Smith and E. Szathmary. *The major transitions in evolution*. Oxford University Press, New York, NY, USA, 1997.
- [112] Elliot Sober and David Sloan Wilson. *Unto Others: The Evolution and Psychology of Unselfish Behavior*. Harvard University Press, 1998.
- [113] Herbert Spencer. *The Principles of Sociology*. Appleton, New York, NY, USA, 1896.
- [114] Kenneth O. Stanley, David D’Ambrosio, and Jason Gauci. A hypercube-based indirect encoding for evolving large-scale neural networks. *Artificial Life*, 2009.
- [115] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.
- [116] Robert Sugden. *The economics of rights, co-operation and welfare*. Oxford: Basil Blackwell, 1986.
- [117] Ron Sun and Dehu Qi. Marlbs: Team cooperation through bidding. In *International Journal of Computational Intelligence Research*, 2005.
- [118] Meyer C. Tanuan. Automated Analysis of Unified Modeling Language (UML) Specifications. Master’s thesis, University of Waterloo, Canada, 2001.
- [119] Adam Tofilski. Influence of age polyethism on longevity of workers in social insects. *Behavioral Ecology and Sociobiology*, 51:234–237, 2002. 10.1007/s00265-001-0429-z.
- [120] Chris Tofts. Algorithms for task allocation in ants.(A study of temporal polyethism: theory). *Bulletin of Mathematical Biology*, 55(5):891–918, 1993.

- [121] James F.A. Traniello and Rebeca B. Rosengaus. Ecology, evolution and division of labour in social insects. *Animal Behaviour*, 53(1):209–213, 1997.
- [122] Sebastian Uchitel, Greg Brunet, and Marsha Chechik. Behaviour model synthesis from properties and scenarios. In *ICSE '07: Proceedings of the 29th International Conference on Software Engineering*, pages 34–43, 2007.
- [123] Sebastian Uchitel, Jeff Kramer, and Jeff Magee. Detecting implied scenarios in message sequence chart specifications. *SIGSOFT Softw. Eng. Notes*, 26(5):74–82, 2001.
- [124] Sebastian Uchitel, Jeff Kramer, and Jeff Magee. Synthesis of behavioral models from scenarios. *IEEE Trans. Softw. Eng.*, 29(2):99–115, 2003.
- [125] Mark VanVugt, Robert Hogan, and Robert B. Kaiser. Leadership, followership, and evolution: Some lessons from the past. *American Psychologist*, 63:182–196, 2008.
- [126] Markus Waibel, Laurent Keller, and Dario Floreano. Genetic team composition and level of selection in the evolution of cooperation. *IEEE Transactions on Evolutionary Computation*, 2009 (To appear).
- [127] Westley Weimer and George C. Necula. Mining temporal specifications for error detection. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 461–476, 2005.
- [128] Jon Whittle and Praveen K. Jayaraman. Generating hierarchical state machines from use case charts. In *14th IEEE International Requirements Engineering Conference (RE'06)*, pages 16–25, Washington, DC, USA, 2006.
- [129] David Sloan Wilson. Weak altruism, strong group selection. *Oikos*, 59(1):135–140, 1990.
- [130] David Sloan Wilson. Altruism and organism: Disentangling the themes of multilevel selection theory. *The American Naturalist*, 150:S122–S134, July 1997.
- [131] Edward 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.
- [132] V.C. Wynne-Edwards. *Animal Dispersion in Relation to Social Behavior*. Oliver & Boyd, London, 1962.
- [133] Jinlin Yang, David Evans, Deepali Bhardwaj, Thirumalesh Bhat, and Manuvir Das. Perracotta: mining temporal API rules from imperfect traces. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, pages 282–291, New York, NY, USA, 2006. ACM.

- [134] Chern H. Yong and Risto Miikkulainen. Cooperative coevolution of multi-agent systems. Technical report, University of Texas at Austin, 2001.
- [135] Ji Zhang and Betty H. C. Cheng. Model-based development of dynamically adaptive software. In *ICSE '06: Proceeding of the 28th international conference on Software engineering*, pages 371–380, New York, NY, USA, 2006. ACM Press.