# ECOLOGICAL EFFECTS ON THE EVOLUTION OF COOPERATIVE BEHAVIORS

By

Brian Dale Connelly

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, and Behavior

2012

# ABSTRACT


ECOLOGICAL EFFECTS ON THE EVOLUTION OF COOPERATIVE BEHAVIORS

By

Brian Dale Connelly

Cooperative behaviors abound in nature and can be observed across the spectrum of life, from humans and primates to bacteria and other microorganisms. A deeper understanding of the forces that shape cooperation can offer key insights into how groups of organisms form and co-exist, how life transitioned to multicellularity, and account for the vast diversity present in ecosystems. This knowledge lends itself to a number of applications, such as understanding animal behavior and engineering cooperative multi-agent systems, and may further help provide a fundamental basis for new industrial and medical treatments targeting communities of cooperating microorganisms.

Although these behaviors are common, how evolution selected for and maintained them remains a difficult question for which several theories have been introduced. These theories, such as inclusive fitness and group selection, generally focus on the fitness costs and benefits of the behavior in question, and are often invoked to examine whether a trait with some predetermined costs and benefits could be maintained as an evolutionarily-stable strategy. Populations, however, do not exist and evolve in a vacuum. The environment in which they find themselves can play a critical role in shaping the types of adaptations that organisms accumulate, since one behavior may be highly beneficial in one environment, yet a hindrance in another. Ever-changing environments further complicate this picture, as maintaining a repertoire of behaviors for surviving in different environments is often costly. In addition to these environmental forces, the number and composition of other organisms with which individuals interact impose additional constraints. The combination of these factors results in significantly more complex dynamics.

Using computational models and microbial populations, this dissertation examines several ways in which ecological factors can affect the evolution of cooperative behaviors. First, envi-

ronmental disturbance is examined, in which a cooperative act enables organisms and their surrounding neighbors to survive a periodic kill event (population bottleneck) of varying severity. Resource availability is then studied, where populations must determine how much resource to allocate to cooperation. Finally, the effect that social structure, which define the patterns of interactions among the individuals in a population, is investigated.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# Chapter 1

# Introduction

Cooperative behaviors are abundant throughout nature. How these behaviors evolved and are maintained, however, has posed a longstanding problem in evolutionary biology. Insight into the evolution and maintenance of cooperation can further our understanding of many features of natural systems, such as the evolution of multicellularity [5], the division of labor [137], and biodiversity, and may even play a critical role in the development of alternative, "anti-infective" treatments to disease [10, 33].

The foundations of cooperation provide benefits that extend beyond the biological world and into the digital realm. Multi-agent systems are becoming increasingly ubiquitous in our world, from sensor networks to schools of robotic fish [119]. As agents acting on behalf of different entities interact with each other, it becomes essential for engineers of such systems to define operating environments where cooperation is encouraged, and where greedy behaviors either serve no benefit or are actively punished. This focus on cooperation has led to the success of the BitTorrent peer-to-peer file sharing protocol [16], where priority is given to those clients that have contributed more. Typically, cooperative behaviors impose some fitness costs on the actor, such as the diversion of limiting resource away from growth [103, 121] or increased visibility to predators [39]. Under the traditional, "survival of the fittest" view of fitness, therefore, organisms displaying such behaviors would not be selected for as often as those that did not incur these costs. Over time,

cooperators would be excluded from the population. Yet, cooperation is ubiquitous. How, then, can the evolution of cooperation be explained?

When the focus of cooperative behaviors is shifted to the level of individual genes, however, selection can favor cooperation. This was the perspective taken by Hamilton, who proposed an expanded view of fitness [55, 56]. Under this "inclusive fitness" theory, which is more thoroughly discussed in Section 2.1, the total fitness of an organism is the combination of that individual's fitness and the fitness of those with whom that organism shares common genes. Therefore, although a cooperative behavior may be costly to the actor, the benefits that are conferred upon the recipient may increase the likelihood for their shared genes to be propagated, which is, evolutionarily speaking, advantageous to the actor. This notion of benefitting other copies of genes was wittily captured by J.B.S. Haldane, who remarked, "I would lay down my life for two brothers or eight cousins." [85]

This expanded view of fitness allows selection for a particular behavior to be considered in terms of its fitness costs, fitness benefits, and the relatedness between the actor and recipient [130]. These parameters are often readily captured in mathematical and computational models of cooperation [2]. Although game theoretic models, further described in Section 2.4, are useful for investigating the roles that various strategies play in a population's ability to evolve and maintain behaviors such as cooperation, they generally do so with a fixed set of costs and benefits. In natural systems, however, the fitness costs and benefits associated with a particular behavior can be extremely difficult or even impossible to measure.

**Thesis Statement:** The evolution of cooperation depends greatly on a variety of ecological factors, including environmental disturbance, resource abundance, and social structure. Variation in any of these factors can alter evolutionary outcomes.

This dissertation describes three ecological factors that have a significant effect on the evolution of cooperation: environmental disturbance, resource abundance, and social structure. As shown throughout this dissertation, these environmental factors continually vary the costs and ben-

efits associated with cooperative behaviors, which in turn affect fitness and ultimately whether or not these behaviors can be maintained in populations. In contrast to previous studies into these factors, the experiments presented in this dissertation focus on evolution in spatially-structured populations, which, as described in the next chapter, create variations in space that can either help or hinder cooperation.

Environmental disturbance is one way in which the environment can affect the evolution of cooperation. Disturbances come in a number of forms and result in the loss of a significant portion of the population. Previous studies have shown that cooperation is favored at intermediate levels of disturbance, when cooperation enables a significant portion of the environment to survive these disturbance events [7]. In Chapter 3, a biofilm model is introduced and used to examine disturbance in spatial populations. Membership in this cooperatively-formed biofilm allows organisms to persist in their environment in the face of varying levels of disturbance.

The abundance of a required resource is another environmental property that has been shown to play a key role in the formation of communities and the types of behaviors they display [121]. As discussed in Chapter 4, resource abundance also plays a critical role in the evolution of cooperation. It is hypothesized that at low levels of resource, the relative costs of cooperative behaviors will simply be too high, and selection will not favor such behaviors. As resource becomes more abundant, though, the relative costs decrease and may reach a point where they are outweighed by the benefits provided. In the studies presented in this chapter, the reciprocal use of both computational and microbial systems to study the evolution of cooperation in environments with different levels of resource produced an alternate interpretation of this relationship. Specifically, populations rapidly transition from not supporting cooperation to supporting it above a critical level of resource.

Finally, chapter 5 presents studies that illuminate the effects of the structure of social interactions on a population's ability to maintain diversity. Diverse populations can be seen as those that can potentially support cooperative behaviors. First, this chapter shows that as interactions within the population increase, oscillations in the fitness of strategies increase to a point where popula-

tions quickly lose diversity. Such increases in interactions could be brought about by increases in population sizes provided by cooperation, as discussed in Chapter 3. The diameter of these interactions within a population, which represents how quickly a dominant strategy can spread, is found to play a determining role in maintaining diversity.

Collectively, these studies enable a more complete understanding of the evolution of cooperation. Most importantly, this body of work demonstrates that evolution does not occur in a vacuum. Rather, its outcomes are shaped by ecological factors that provide selection with a continuously moving target. These results signal that care should be taken to account for such factors when studying the evolution of cooperation. Through the use of computational and microbial models, this work also offers new interpretations of how two major ecological factors—population bottlenecks and resource abundance—affect the evolution of cooperation. Additionally, this work demonstrates that the structures of the networks of interactions within populations are just as important in determining evolutionary outcomes as more commonly used indicators such as population size. Further, this work shows that the introduction of novelty through mutation, migration, or even drift allows populations to more quickly adapt as selection alternates between favoring and not favoring cooperation. When these fluctuations occur, novelty reduces the magnitude of fitness changes, which in turn can prevent the loss of cooperation.

These investigations also demonstrate the contributions that computer science and computational models can make towards understanding fundamental properties of biology and science in general. Perhaps nowhere is this more apparent than in the study of evolution, which can be difficult to study due to the long timescales on which it acts. Through the use of computational models, populations can easily be studied for thousands or millions of generations. Another benefit of computational models is the level of experimental control that they provide. With computational models, researchers have complete control over the individuals studied and the environments in which they exist. Moreover, computational systems offer complete insight into each individual and its behaviors, which is impossible in other experimental systems.

As with any endeavor, success depends on choosing the right tool for the job. Microbial, math-

ematical, and computational systems, which are commonly used to study evolution and ecology, each have inherent strengths and weaknesses. This work benefited tremendously from the reciprocal use of computational and microbial systems, which enabled one system to inform the other, which produced new insights that were incorporated and further examined. Feedback loops such as these can profoundly increase both the breadth and depth of research, and the *consilience* of results gathered this way can greatly increase confidence [131].

We begin with an introduction of the concepts and techniques upon which this work is based, followed by chapters describing each of the three main topics of this work: environmental disturbance, resource abundance, and social structure.

# Chapter 2

# Background

This chapter introduces some of the key theory and techniques used throughout the remainder of this document. Section 2.1 presents Hamilton's rule and inclusive fitness theory, a gene-centric view of the economics associated with a given behavior. The conditions laid forth by Hamilton's rule, which describes how evolution can favor costly cooperative behaviors, are the basis for the design of each of the experiments detailed in the remaining chapters. One such cooperative behavior that is modeled throughout this document is biofilm formation, which is introduced in Section 2.2. Section 2.3 discusses how spatial structure and localized interactions can play key roles in determining whether or not cooperation can be favored. Game theory is described in Section 2.4. Although game theoretic methods are not used for any of the experiments described in this document, they are frequently used in related work, and their limitations provide inspiration for this body of work. Cellular automaton models and the SEEDS platform for studying evolutionary and ecological dynamics using these models are introduced in Sections 2.5 and 2.6. Finally, the Avida digital evolution platform is presented in Section 2.7. Section 2.8 synthesizes these concepts and shows how they motivate and enable the work presented in the following chapters.

## 2.1 Inclusive Fitness and Hamilton's Rule

How cooperative behaviors can evolve has long been considered a difficult problem in evolutionary biology. The selection for costly behaviors instead of those that do not incur those costs puzzled even Darwin, who wrote, "If it could be proved that any part of the structure of any one species had been formed for the exclusive good of another species, it would annihilate my theory, for such could not have been produced through natural selection." [29]

Frequently, cooperative behaviors result in the production of *public goods*, which are common resources available to all members of a group. Public goods comprise a wide variety of products, including siderophores, which offer increased access to iron for populations of *Pseudomonas aeruginosa*, and exopolysaccharides, which allow populations of many species of microorganisms to form biofilms, the benefits of which are further discussed in Chapter 3.

These public goods are subject to the *Tragedy of the Commons*. Garrett Hardin famously described this phenomenon as a common pasture shared among farmers, where it was in each farmer's best interest to add as many cattle as possible. The tragedy arises, therefore, because the costs of adding these cattle are incurred by all farmers, which quickly decrease the value of the shared resource and lead to the demise of the pasture [60]. A great number of studies have focused on how public goods cooperation can be robust in the presence of cheaters [111].

Although several theories, such as group selection [132, 136], have been presented to explain how cooperative behaviors can evolve and resist invasion from *cheaters*, perhaps none has been as well adopted as *inclusive fitness* and *kin selection*, which were introduced by W.D. Hamilton in 1964 [55, 56]. With his theory, Hamilton shifted the focus of selection from the individual to the genes that are shared by the individual and its relatives. This "inclusive" fitness allows for behaviors to be favored as long as the fitness benefits provided by that behavior, when observed among all individuals with the gene in question, outweigh the fitness costs of the act. This relationship, often referred to as Hamilton's rule, is shown in Equation 2.1.

$$C < r * B \tag{2.1}$$

In this inequality, $C$ represents the reproductive cost incurred by the individual performing the cooperative act, $B$ represents the reproductive benefits awarded to the recipients, and $r$ is the relatedness between the actor and recipient. Originally, Hamilton used Sewall Wright's coefficient of relatedness as a measure for $r$, which is defined as the probability that the alleles at a random locus would be identical by descent [133]. Later invocations of this theory would instead use a regression coefficient comparing the relatedness between two interacting organisms to the average relatedness in the population, which more readily allows for selection of spite and other negative interactions [54, 106]. Behind the three terms of Hamilton's rule lie a sizeable number of complexities [44], and several extensions have been proposed that aim to more clearly identify the different types of interactions present in populations, e.g. [108].

Although it is tempting and perhaps natural to consider the costs and benefits of cooperation at the level of the individual, care should be taken to remember that Hamilton's rule works from a gene-centric perspective. That is, a behavior that costs an individual more than the benefit that the individual receives will still be favored if that behavior benefits the genes in question more than its costs.

Selection for cooperative behaviors, therefore, requires insights into the fitness costs and benefits of the behaviors, as well as the relatedness among organisms in the populations. One way in which high relatedness can be maintained among interacting individuals is through localized interactions, a topic which is further discussed in Section 2.3. Alternately, a number of mechanisms whereby organisms can interact preferentially with close relatives have been identified [118].

So what determines the fitness costs and benefits of a behavior? The work presented in this dissertation maintains that environmental conditions play a major role in determining both the costs of a behavior and the significance of the benefits that it provides. If so, selection poses an ever-moving target for these behaviors, as environments can change dramatically though a myriad of processes. By examining populations over longer time periods in these dynamic environments, we can gain valuable insights into exactly how the costs and benefits of cooperative behaviors change and drive selection.

## 2.2 Microbial Biofilms

Biofilms are aggregates of microorganisms, typically bacteria or yeast, connected within a matrix of extracellular polymeric substance (EPS), extracellular DNA, and additional proteins [25, 26]. Bacteria are now believed to spend the majority of their lives in biofilms, as opposed to the planktonic, or free-swimming state [25]. In contrast to the traditional view of bacteria as isolated individuals, this new understanding has begun to shed light on the vast repertoire of intra- and inter-species social behaviors exhibited in this biofilm lifestyle [129] as well as the complexities of these colonies [31].

The production of EPS by cells is a cooperative behavior, and inclusion in the resulting biofilm provides substantial benefits. Importantly, biofilms enable their constituents to maintain spatial structuring, even in well-mixed environments [25, 110]. As discussed in the next section, spatial structuring can be a major factor in the evolution of cooperation. Also, the "stickiness" of this matrix allows organisms in biofilms to cohabitate in larger densities and adhere to surfaces, enabling those organisms to resist being flushed out of their environment [53]. In addition, the structure and adhesion provided by biofilms can offer greater access to resource [110]. In addition biofilms play a major role in antibiotic resistance, and can render bacteria up to 1,000 times less likely to be affected by antibiotics [116] or other antimicrobial agents [82].

Although biofilms provide constituents with a number of benefits, they often pose a number of threats when they allow colonization of human tissue [53] and other surfaces. Perhaps the most well-known biofilms are those that form dental plaque [83], which is associated with a number of oral and general health concerns. Highly-resistant *Pseudomonas aeruginosa* biofilms chronically infect the lungs of cystic fibrosis patients, where they cause pneumonia and a host of other problems [26]. Biofilms also frequently infect catheters and prosthetic devices, greatly diminishing the efficacy of those devices [53]. A greater understanding of how cooperation enables the creation and persistence of biofilms could offer new avenues for treatments of these and other problems [4, 33, 112].

The experiments presented in Chapters 3 and 4 use a model of cooperation inspired by biofilm

formation. In these models, cooperators produce a public good that better enables them to survive in a harsh environment. This is akin to the production of EPS in the formation of biofilms, which enables adherence to surfaces and resistance to being flushed out of the environment. Like their natural counterparts, these "virtual biofilms" present an opportunity for exploitation by cheaters, who take advantage of the benefits provided by the biofilm without contributing to its formation.

## 2.3 The Importance of Spatial Structure

Spatial structuring has frequently been shown to play a decisive role in the ability of a population to evolve and maintain cooperation [72, 87, 122]. When spatial structuring exists, an organism's interactions are limited to others that are located nearby. If dispersal is limited, patches of relatives form as offspring are placed into the surrounding environment. This pattern of growth makes an organism more likely to interact with a close relative than mere chance would allow. Such localized interactions also mean that the benefits of cooperation are more likely to affect relatives. Both of these factors can increase the likelihood of satisfying Hamilton's rule.

As interactions persist, one of the primary disadvantages that can arise as a result of localized interactions, however, is an increase in competition among kin [75, 104]. This occurs when successful cooperative groups grow large enough that the benefits, when divided among that group's constituents, become overly diluted. It is at this point when additional behaviors may become necessary to maintain cooperation. One frequently-observed reaction to increases in kin competition is dispersal, where some members of the group emigrate to new areas where competition among kin is potentially lower [75, 104].

As organisms associate in space, they also alter their environment, both positively and negatively, through *niche construction* [32, 76, 94]. Because of the resulting spatial variations in phenotypes and environmental conditions, selection will act differently from patch to patch. Queller [107] has shown that when this is the case, Hamilton's rule still holds true when considering the relatedness within patches, instead of across the entire population.

Spatial structuring is a core theme of this dissertation. The results presented in Chapter 3 extend support for spatial structuring's role in enabling cooperation, showing that localized, persistent interactions result in larger and more robust populations of cooperators than when interactions occur in well-mixed populations. Based on these results, spatial populations were studied in all subsequent computational models. Chapter 5 directly addresses the effect that different kinds of spatial structuring have on the ability to maintain diversity and cooperative behaviors in populations, and demonstrates that the overall structure of the interactions, particularly how the amount of time that takes for a dominant strategy to spread in a population through interactions can determine whether or not cooperation can persist.

## 2.4 Game Theoretic Models of Cooperation

Game theory [113] has frequently been applied to study cooperation and other collective behaviors in populations [37]. Game theoretic models typically consist of a series of pairings between two individuals selected from the population. These pairings can be randomly selected or chosen from a local neighborhood, and each pairing may occur once or in an "iterated" fashion, where the players play more than once in succession and may remember some portion of the strategies used in previous rounds. Game theoretic models have provided insights into the dynamics of social behaviors, such as the effects of spatial structures [92], dynamic social ties [102], and nonlinear benefits and resources [47].

The results of these pairings on the players depends on the combination of strategies played, which are traditionally defined in a *payoff matrix* before the onset of the experiment. Table 2.1 shows the format commonly used to display such payoff matrices, where rows represent the strategies played by Player A, and columns represent the strategies played by Player B. The cells where they intersect represent the payoff that Player A and Player B would receive, respectively, if they played those strategies when paired. Here **R** represents the reward for mutual cooperation, **T** represents the temptation to defect when the opponent cooperates, **S** is the sucker's payoff when

cooperation is played while the opponent defects, and **P** is the punishment for mutual defection.

Table 2.1: Structure of a Payoff Matrix

| | Cooperate | Defect |
|---|---|---|
| Cooperate | R,R | S,T |
| Defect | T,S | P,P |

Game theoretic models are typically differentiated by the relationship between the values in these payoff matrices. One of the most well-known models used is *Prisoner's Dilemma* (PD) [2], in which $\mathbf{T} > \mathbf{R} > \mathbf{P} > \mathbf{S}$. *Snowdrift*, also known as *Hawk-Dove* [114] or *chicken*, is another game theoretic model frequently used to study the spread and maintenance of cooperative behaviors in populations [38]. Importantly, the relative payoffs for Snowdrift differ from those used in PD in that $\mathbf{T} > \mathbf{R} > \mathbf{S} > \mathbf{P}$. Because of this, these models are understood to have different dynamics in the evolution of cooperation [62].

Game theoretic models have produced a number of key insights into cooperative behaviors [93], even though the values commonly used in payoff matrices impose two critical limitations. First, fitness costs and benefits can be difficult to measure in natural systems, so determining biologically-accurate values to be used in payoff matrices may pose an insurmountable challenge. Secondly, the payoff matrices typically contain fixed, *a priori* values and predefined relationships between those values. This creates a relatively closed environment, and any behaviors or population compositions will evolve to match this specific environment, and will not necessarily apply in others. Further, these populations cannot affect their environment through their behaviors. Although factors such as density dependence and initial conditions allow populations to explore different trajectories in game theoretic models, the lack of environmental variation may alter selection in such a way that the dynamics of the experiments do not match those present in natural populations. Because of this, models such as cellular automata and Avida, presented in the following sections, may be more appropriate for studying the evolution of cooperation, particularly in dynamic environments.

## 2.5 Cellular Automata

Cellular automata are another model frequently used to study the evolution of cooperative and other behaviors in spatial environments is the cellular automaton (CA) [28, 71, 125]. In these models, individuals reside in the cells of a finite-sized lattice, and their interactions are limited to their neighboring cells. Typically, neighborhood sizes of 4 (Von Neumann) or 8 (Moore) cells are used; however, the specific topology of the world can follow other forms. When the strategies being observed are density dependent, the size of an individuals' neighborhood can have a significant effect on the behaviors exhibited by those individuals, as will demonstrated in Chapter 5.

In a cellular automaton, at each unit of time, often called an *epoch*, cells are updated based on the interactions they have with neighboring cells. These updates may cause an individual residing in that cell to die or to adopt the strategy of a neighboring cell. The specific update rules are determined by the question being asked. For example, an individual that uses quorum sensing as part of its strategy may be more likely to die (or less likely to reproduce) at a given update, which would represent the additional costs of this behavior. Updates can be *synchronous*, where the order of cells updated follows some defined pattern, and all updates are based on the state of the cells at that time. Alternatively, the population can be updated *asynchronously*, where cells are updated in random order, and the state of a cell at time $t$ may depend on the state of a neighbor at either time $t$ or $t+1$.

Since cellular automaton models often have a finite number of cells in which individuals can reside, space becomes a limiting resource [121]. Indeed, in such models, space often serves as the only resource for which individuals contend. Those individuals that outcompete their neighbors are rewarded with more cells, allowing their strategy to reach greater densities. Once a particular strategy has successfully excluded all other strategies and occupies all cells, no other strategies will be observed unless those strategies can re-emerge through mutation or migration. These topics of spatial limitation and the re-emergence of strategies in CA models will be discussed in detail in Chapters 3 and 5.

One of the primary shortcomings of cellular automata is that the set of behaviors an individual

can display are often predefined. Although mutations may alter an individual's investment in a particular behavior, novel behaviors are not likely to be seen. Further, the mapping between an individual's genotype and phenotype is likely to be direct, so an individual's behavior is not likely to change in response to the state of its neighborhood, the environment, or stochastic processes, unless they are programmed to do so in some predetermined way.

Despite these shortcomings, this technique can offer tremendous insight into the evolution of behaviors and their stability in spatial worlds. By observing the abundances of different strategies over time, the evolutionary trajectories and stabilities of those strategies can be estimated. This makes them ideal for the work presented in Chapters 3 and 5, where the relationship is examined between the social structure of a population and whether or not that population can maintain diverse strategies. The SEEDS platform, upon which these models are built, is described in the next section.

## 2.6 The SEEDS Platform

SEEDS is a software platform designed to facilitate the creation and use of cellular automaton models to conduct experiments targeting questions in ecology and evolutionary biology. Although initially developed to study the role that social structure plays in the maintenance of diversity, which will be presented in Chapter 5, it has since been used to explore a variety of topics, such as the use and maintenance of horizontal gene transfer [22], the evolution of cooperation in probabilistic game theoretic models, and speciation and sexual dimorphism resulting from divergent selection in limited resource environments. Additionally, SEEDS was used to create one of the models of bacterial growth and cooperative biofilm formation described in Chapter 3.

Model populations in SEEDS consist of a set of cells connected by an arbitrary graph. Each cell represents an individual, and edges connecting cells in the graph represent potential interactions between those cells. The use of arbitrary graphs allows experiments to capture population structures ranging from well-mixed populations to single, structured populations to metapopula-

tions consisting of many subpopulations either existing in isolation or connected by migration. Further, the structure of these populations can change over time as populations grow and affect their environment.

SEEDS provides a number of building blocks for these populations, which users combine to best represent the problem at hand. Typically, the user creates a new cell type, which defines the properties and behaviors of each individual. For example, each cell could represent a state, such as alive or dead, as in Conway's Game of Life [46] and other traditional CAs. Alternately, each cell could encapsulate an artificial neural network, such as those evolved using NEAT [115]. In the models described in Chapters 3 and 5, cells were created that modeled cooperative biofilm production and the production of antibiotics by bacteria, respectively. By adding additional complexity, a cell could even represent a computer program and have virtual hardware associated with it, as in the Avida system [95], which is presented in the next section. Appendix A provides a thorough description of the SEEDS platform, including its design, use, and extension.

## 2.7 The Avida Digital Evolution Platform

Avida [95] is a relatively-complex cellular automaton system that has been used to study a wide variety of complex ecological and evolutionary systems, such as adaptive radiation [14], the evolution of complex features [80], systems of communication [73], and cooperative navigation [20]. Compared to many CA systems, Avida enables relatively open-ended evolution. Avida was used for the experiments presented in Chapters 3 and 4. Because Avida is a highly-configurable system, this section describes Avida as it was used in the models presented in this dissertation.

Experiments in Avida focus on populations of self-replicating digital organisms, each of which resides independently in a cell on a lattice. As depicted in Figure 2.1, each organism, or *Avidian*, in this system consists of a circular list of assembly-like instructions (its "genome"), which are executed sequentially on virtual hardware allocated to that organism. This virtual hardware comprises three 32-bit registers, two stacks, and a CPU, which has four heads that control the execution

flow of the organism's genome and aids in self-replication. The 26-instruction instruction set available to organisms is Turing complete, so organisms can feasibly perform any possible calculation. Through the use of jumps and other instructions, organisms may alter their behaviors throughout their lifetimes, which is an enormous benefit offered by this system. More explicitly, one genotype may produce several phenotypes.

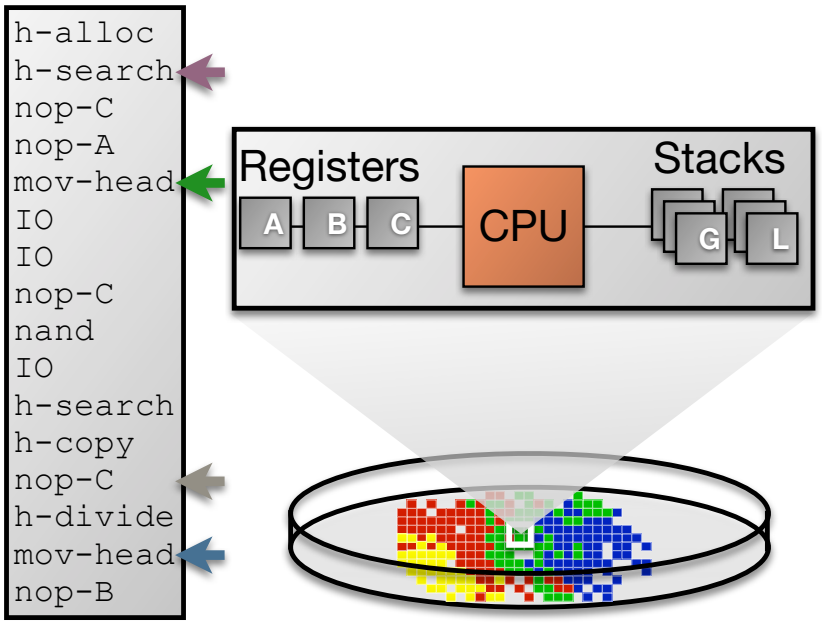



Figure 2.1: In Avida, each individual organism (top) in a population (below) consists of a genome (left), which is run on virtual hardware associated with that organism. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

An organism replicates by first allocating additional space in its genome in which to create a copy of itself. It then copies instructions line by line from its genome into this new area. As an organism copies its genome, point mutations cause instructions to be modified with a configured probability. As the organism divides, mutations may also cause instructions to be added or removed from the genome. Once this process has completed, the genome is cleaved, and the new organism is placed into a randomly-chosen neighbor cell, replacing any existing organism residing in that cell. Because a parent is adjacent to its offspring in this model, dispersal is limited, and spatial structure is maintained. As an alternative, well-mixed populations can be studied in which offspring are placed into random cells in the environment.

Organisms compete for space in this environment by completing logic *tasks* [80], which confer a *merit* bonus to the organism's offspring. An organism with more merit receives more virtual CPU cycles, enabling it to execute its genome faster and thus spread more quickly in the environment. During each *update*, Avida's unit of time, an organism executes an average of 30 instructions. Those with higher merit execute proportionally more instructions per update, and those with lower merit execute proportionally fewer instructions.

In order to complete a task, an organism must first execute the `IO` instruction, which places randomly-generated numbers into its registers. The organism is then required to execute the instructions necessary to complete the task, place the result in the correct output register, and issue an additional `IO` instruction. A full list of the tasks available by default and their rewards is given in Table 2.2. These tasks may have varying metabolic costs, or the number of instructions required for their completion, as is shown in Figure 2.2.

Table 2.2: Logic tasks that can be completed by organisms and the merit rewards they confer

| Task | Input | Output | Merit Bonus |
|---|---|---|---|
| `NOT` | $A$ | $\neg A$ | 2 |
| `NAND` | $A, B$ | $\neg(A \wedge B)$ | 2 |
| `AND` | $A, B$ | $A \wedge B$ | 4 |
| `OR NOT` | $A, B$ | $A \vee \neg B, \neg A \vee B$ | 4 |
| `OR` | $A, B$ | $A \vee B$ | 8 |
| `AND NOT` | $A, B$ | $A \wedge \neg B, \neg A \wedge B$ | 8 |
| `NOR` | $A, B$ | $\neg(A \vee B)$ | 16 |
| `XOR` | $A, B$ | $(A \wedge \neg B) \vee (\neg A \wedge B)$ | 16 |
| `EQU` | $A, B$ | $(A \wedge B) \vee (\neg A \wedge \neg B)$ | 32 |

Although the finite amount of space serves as an important limiting resource in Avida, additional resources can also be defined. These resources can flow into and out of the environment, as well as decay at configured rates. Further, resources can be required for the completion of tasks, as is done in Chapter 4, or be created as a byproduct of tasks.

Typically, populations are seeded with a single organism capable only of replication. Any additional behaviors are therefore evolved through mutations introduced during the replication process. Usually, several populations are evolved in parallel, but with different seeds to the pseudorandom

**(a)**
```
IO
IO
nop-c
nand
IO
```

**(b)**
```
IO
IO
nop-c
nand
nand
IO
```

**(c)**
```
IO
IO
nop-c
push
nand
swap
nand
swap
nop-a
pop
nand
swap
nop-c
nand
IO
```

**(d)**
```
IO
IO
nop-c
push
nand
swap
nand
swap
nop-a
pop
nand
swap
nop-c
nand
push
pop
nop-c
nand
IO
```

Figure 2.2: Sample genome segments required to complete several tasks. (a) The `NAND` task can be completed with 5 instructions, while (b) `OR NOT` requires 6. More difficult tasks such as (c) `XOR` and (d) `EQU` require at least 15 and 19 instructions, respectively.

number generator, which allows these populations to explore different evolutionary paths.

In the following chapters, a virtual biofilm model is developed using Avida. Through this model, cooperation is studied as it relates to environmental disturbance and resource availability. In these studies, Avida's open-ended nature allows organisms to evolve a wide variety of competitive- and cooperative phenotypes.

## 2.8 Toward Understanding Ecological Effects on the Evolution of Cooperation

The previous sections have introduced the topics that underlie the work described in the remaining chapters. First, Chapter 3 develops and uses models of cooperative biofilm formation to show that the level of disturbance present in an environment affects the evolution of cooperation in

both positive and negative ways. In these models, which were created using Avida and SEEDS, cooperators are able to survive disturbance through the production of biofilms.

This *virtual biofilm model* is further used in conjunction with microbial populations in Chapter 4, where the availability of a required resource is shown to have a strong effect on evolutionary outcomes. In populations of both *Vibrio cholerae* and digital organisms, cooperation is shown to be favored once resource abundance crosses a critical threshold, thus satisfying Hamilton's rule.

Finally, Chapter 5 demonstrates the strong effect that spatial structuring has on the ability for populations to maintain diversity and cooperative toxin production. Here, the use of different graph structures in the SEEDS platform enabled experiments to study populations with a variety of social and spatial relationships. In situations where spatial structure was unable to allow diversity to be maintained, the introduction of novelty through either mutation or migration is shown to promote diversity.

# Chapter 3

# Environmental Disturbance

Environmental disturbance has long been known to affect the composition of communities living in a particular environment. Disturbance can come from a number of sources, including the flow of liquid through an environment, grazing by higher organisms, and even antibiotic treatment. The *Intermediate Disturbance Hypothesis* (IDH) [18, 50] postulates that the potential for diversity in a population is maximized when intermediate levels of disturbance are present in the environment. Such an environment can potentially allow the coexistence of species that invest solely in growth and those that compete through costly behaviors.

Cooperation is one way in which a species can invest in its competitive ability, and such behaviors can be greatly affected by the amount of disturbance present in the environment. The effect of the environment on cooperation has previously been studied in the laboratory using populations of *Pseudomonas fluorescens*, where cooperation was indeed found to peak at intermediate levels of disturbance [7]. At low disturbance, the benefits of cooperation, increases in survival provided by biofilm formation in this case, provided insufficient fitness benefit to outweigh the cost. Similarly, in high-disturbance environments, membership in a biofilm simply could not provide enough protection, and the benefits once again did not outweigh the costs. However, at intermediate disturbance, the adhesion provided by the biofilm benefited populations enough to outweigh the costs of production, and thus cooperation was maintained.

The work presented in this chapter examines how environmental disturbance affects cooperation in spatially-structured populations. Specifically, this work investigates the conditions under which cooperation can emerge, how prevalent cooperation is within populations, and how the level of disturbance affects both the amount of cooperation and the composition of the population.

Section 3.1 describes how Avida was extended to create a computational model of bacterial biofilms, while Section 3.2 examines populations in these environments. Studies of how populations of cooperators react when disturbance is increased are discussed in Section 3.3, and how these model populations react when disturbance is lost and later re-introduced is presented in Section 3.4. Finally, Section 3.5 examines how population bottlenecks, which are large disturbance events, affect the evolution of cooperation. The results of these experiments demonstrate that disturbance from the environment can be a driving force in the social lives of natural organisms.

## 3.1 Virtual Biofilm Model

One area in which cooperative behaviors are critical is in the formation and maintenance of biofilms [89], introduced in Section 2.2, which are aggregates of microorganisms connected within a matrix of extracellular polymeric substance (EPS) [26]. Among many other benefits, biofilms adhere to surfaces, enabling their constituent organisms to resist being flushed out of the environment. Biofilms also promote spatial structuring, a critical component of cooperation discussed in Section 2.3, even in well-mixed environments [110].

This self-produced EPS is a public good in this system, which creates the potential for organisms to take advantage of its benefits without producing it themselves. These cheaters would then have a selective advantage over cooperators. The presence of such cheaters has been observed to significantly weaken biofilms [110].

To model the ecological and evolutionary forces faced by populations of bacterial biofilms, the Avida digital evolution platform was extended so that organisms could evolve to form virtual biofilms [20]. As in nature, an organism's membership in a virtual biofilm provided it with some

benefit that could not easily be obtained in a free or planktonic state. In this environment, inclusion in a biofilm allowed organisms to survive a periodic "kill" event.

To create a virtual biofilm, organisms completed the `OR NOT` task, which resulted in one unit of a public good being placed into the environment at their cell. This cooperative resource can be likened to EPS, enzymes, or other public goods commonly observed in microbial populations. Unlike the other available tasks, `OR NOT` did not confer a merit reward; however, the public good that it produced could enable that organism and its neighbors to survive a disturbance, as discussed below. Cooperators in this model are therefore defined as organisms who performed this task at some point during their lifetimes, while cheaters focused solely on tasks that garnered individual fitness rewards.

To introduce disturbance, a kill event periodically selected a focal cell at random, and examined each cell in a neighborhood comprising all cells within a configured distance from that focal cell. If the average level of public good in that neighborhood was below a configured threshold, organisms residing in those cells were killed. This kill event, `KillWithinRadiusMeanBelowResourceThreshold`, and several variants have been included in Avida distribution.

The public good both decayed and diffused into neighboring cells at each update. In this environment, cheaters could exploit the diffusion of public good from neighboring cooperator cells while refraining from producing the public good themselves. Further, the decay of this resource ensured that the cooperative act would need to persist in order to maintain protection from the kill event. To allow for populations to first reach a sufficient density, the kill event began at update 1,000 and occurred during each subsequent update.

Because each logic task in Avida can potentially serve as either a building block or a hindrance to completing another task, as previously observed [80], populations were first evolved in an environment without the periodic kill event in order to determine how many `OR NOT` tasks would be completed (and consequently how much public good would be produced) when this behavior was neither rewarded nor improved an organism's chances of survival. These populations produced a

mean per-cell resource level of 0.37 units. By using a significantly higher threshold in the virtual biofilm experiments, such as the threshold of 3.0 as used in the experiments described in this section, it can be inferred that the success of organisms at staving off periodic killing was not simply a byproduct of completing rewarded tasks.

The costs of cooperation in this model are related to the metabolic effort required to complete the cooperative task enough times to provide resistance to the kill event instead of focusing on the rewarded tasks. Because of the diffusion of public good, these costs could be incurred by- or distributed among neighboring organisms. The benefits of cooperation are increased survival of the kill event when sufficient public good has been produced. Quantifying these costs and benefits, therefore, depends on the composition of cooperators in an organism's neighborhood, the efficiency with which the cooperative task is completed, the decay and diffusion of the public good, as well as the frequency and size of the kill event. Because of these forces, the costs and benefits of cooperation vary over time and space. Although no resources are required to complete tasks in this model, space in this world is limited, which creates a pressure to grow quickly to outcompete neighboring cells. This, in turn, creates selective pressure to complete rewarded tasks and to stay alive and reproduce. Whether or not the cooperative phenotype is favored can depend greatly on how selection acts in that environment. The following sections describe a set of experiments where the strength of disturbance in this model was controlled event to examine how environmental disturbance affects the evolution of cooperation.

## 3.2 Spatial Effects

First, the role that spatial structuring plays in the evolution of cooperation is demonstrated using this virtual biofilm model. Table 3.1 lists the parameter values for experiments described in this section. Populations were evolved for 100,000 updates, or approximately 17,680 generations in either spatially-structured or well-mixed environments. In spatially-structured populations, offspring were placed into neighboring cells. In well-mixed populations, these offspring were instead

placed into a randomly-chosen cell in the environment.

Table 3.1: Virtual Biofilm Model Parameters for Spatial Effects Experiments

| Property | Value |
|---|---|
| Population Size | 10,000 |
| Replicate Populations | 50 |
| Updates | 100,000 (17,680.47 $\pm$ SEM 521.91 generations) |
| Kill Event Start Update | 1,000 updates |
| Kill Event Frequency | 1 update |
| Kill Event Radius | 5 |
| Kill Event Focal Cells Per Update | 1 |
| Kill Event Public Good Threshold | 3 units |
| Public Good Diffusion Rate | 0.01 |
| Public Good Decay Rate | 0.01 |
| Copy Mutation Probability | 0.0075 |
| Insertion Mutation Probability | 0.05 |
| Deletion Mutation Probability | 0.05 |

The abundance of cooperators and cheaters in spatial populations are shown in Figure 3.1. At the end of these experiments, an average of approximately 48% of occupied cells were cooperators (public good producers). Cooperators are defined as those individuals which had completed the public-good-producing task. Cheaters, which focused solely on the completion of one or more rewarded task, and not producing the resource, accounted for 52%. As shown in Figure 3.3, cooperators and cheaters formed patches in these populations. These patches helped cooperators target the benefits of public good production towards relatives as well as avoid invasion by cheaters. Although only about half of the living organisms in the populations produced public good, it was sufficient to prevent 67% of selected organisms from being killed, as shown in Figure 3.2.

The level of resource in each cell was also measured as the experiments progressed. One might predict this level to converge to the threshold amount of 3 units; however, modest stability was reached at a per-cell average of approximately 6.6 units, as shown in Figure 3.4. This surplus can be viewed as cooperative: Cooperators produced enough public good to enable themselves to survive and to help their neighbors survive as well. This asymmetry follows the *Tragedy of the Commune* [38], where levels of investment in public goods may not be uniform.

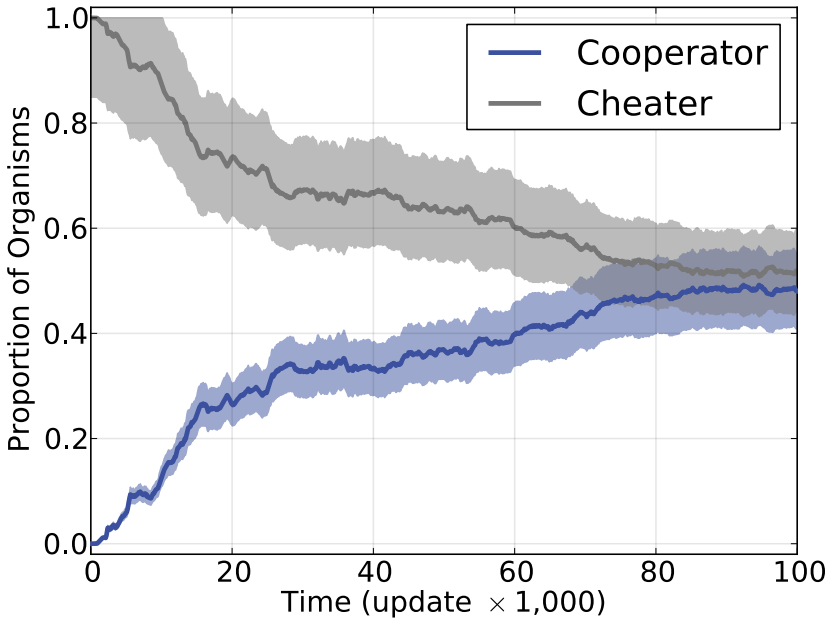



Figure 3.1: Distribution of behaviors within spatially-structured populations (n=44). In these experiments, the benefits provided by cooperation allowed cooperators to reach the same densities as cheaters. Shaded regions indicate $\pm 1$ standard error.

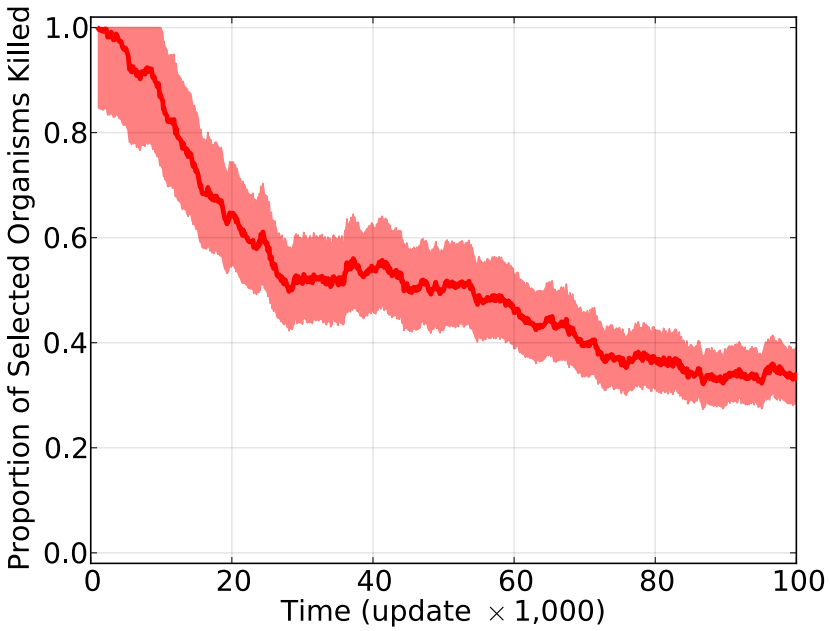



Figure 3.2: Fraction of selected organisms killed in spatially-structured populations (n=44). Shaded region indicates $\pm 1$ standard error.

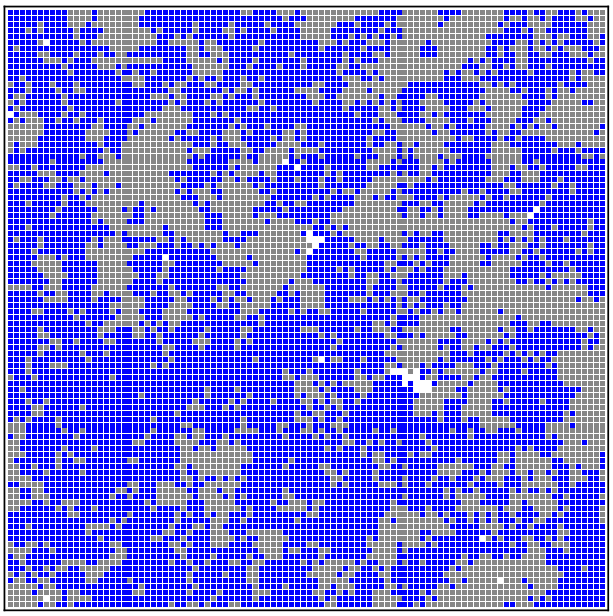


Figure 3.3: Snapshot of one replicate spatially-structured population containing patches of cooperators (blue), cheaters (grey), and empty cells (white)

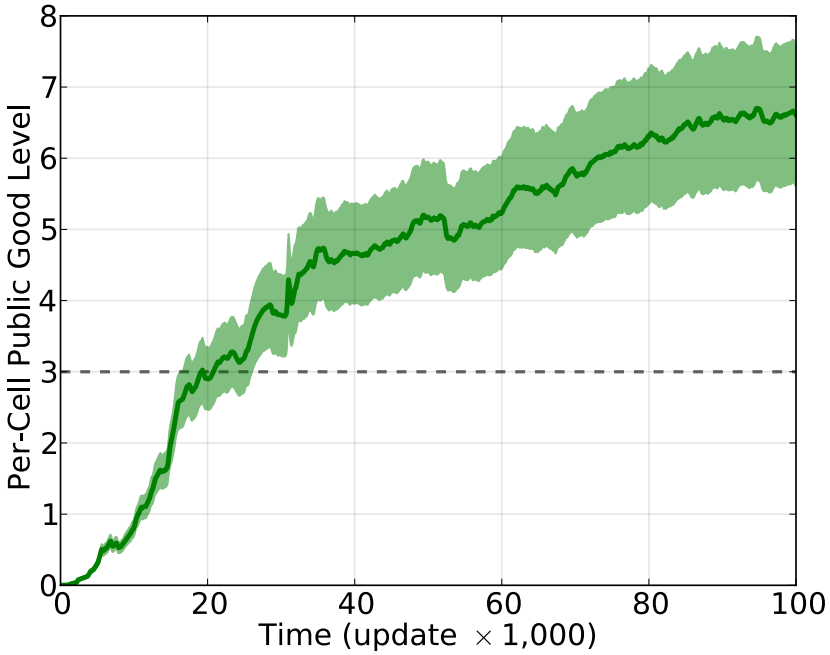



Figure 3.4: Average per-cell public good levels in spatially-structured populations (n=44). Although these populations produced levels that were considerably above the kill threshold of 3 units, they represent an average over space. Public good levels in areas near patches of cooperators were higher than this, while areas near patches of cheaters were lower than this. Shaded region indicates $\pm 1$ standard error.

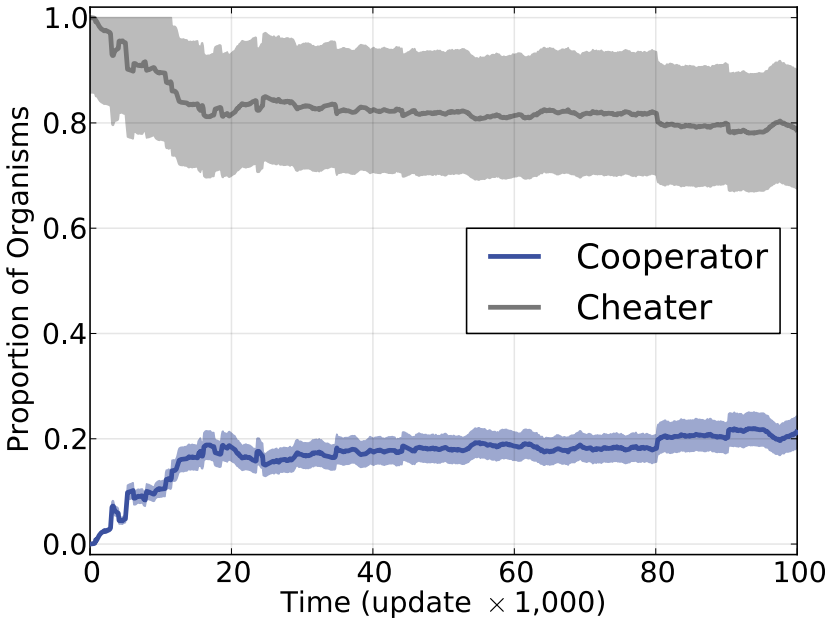
In well-mixed populations, individuals are far less likely to remain near relatives. As Figures 3.5a and 3.6 show, although cooperators were able to evolve in this environment, they were far less abundant in the populations. Because of this, these populations were considerably more susceptible to disturbance. Figure 3.5b shows that on average approximately 77% of those cells examined by the kill event were killed. Because cooperators were unable to reach a substantial abundance in most populations, the average per-cell level of public good among well-mixed populations was 1.02 units, as shown in Figure 3.7. Interestingly, however, cooperators were able to maintain a large proportion of the population in 12 of the 48 replicate populations. In these instances, the public good allowed cooperators to survive the kill event and reproduce, replacing randomly-chosen cheater cells.

As shown in Figure 3.8, spatial populations are much more likely to allow for the evolution of cooperation. This is because the cooperative production of public good is often far more costly than the benefits it provides, because these benefits are more likely to affect cheaters more than cooperators. Based on these results, spatially-structured populations are used for the remainder of the experiments discussed in this dissertation.

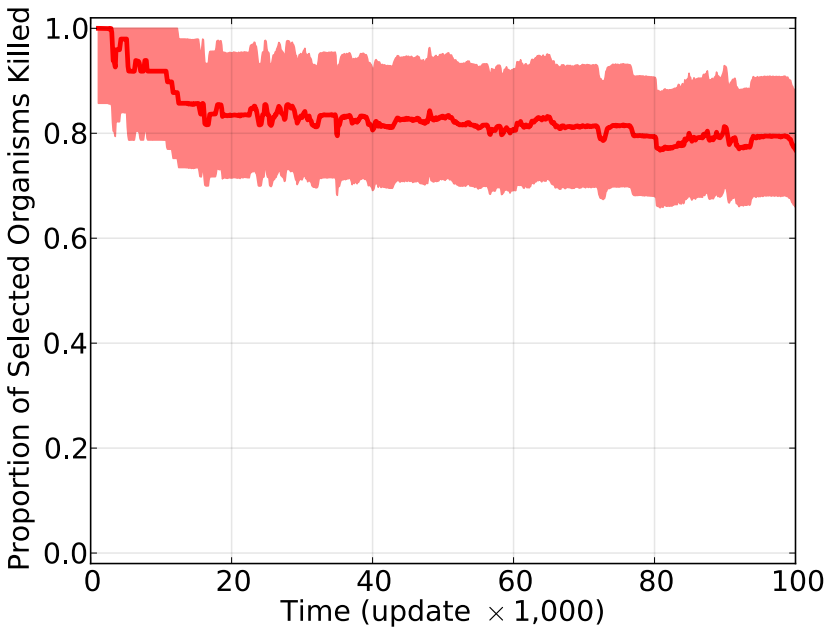
In these experiments, the level of disturbance imposed by the kill event was intermediate. Cooperation was able to persist in spatially-structured populations, because the production of public good better enabled cooperators to grow and survive in these environments, even in the presence of cheaters. As the levels of disturbance increase, however, these benefits might not be enough to overcome this adversity, and selfish growth may once again be a favorable strategy if it allows individuals to grow at a rate that outpaces the damages inflicted by the kill event. In the next section, the effect of increased disturbance is addressed.

## 3.3 Increasing Disturbance

To determine how the level of disturbance affected the evolution of cooperation, kill events with radii between 6 and 13 cells were tested. Because the strength of the kill event was controlled




(a)




(b)

Figure 3.5: Cooperators and cheaters in well-mixed environments: (a) Proportions of cooperators and cheaters among 40 populations (b) Fraction of selected organisms killed within a 5-cell radius. Shaded regions indicate $\pm 1$ standard error.

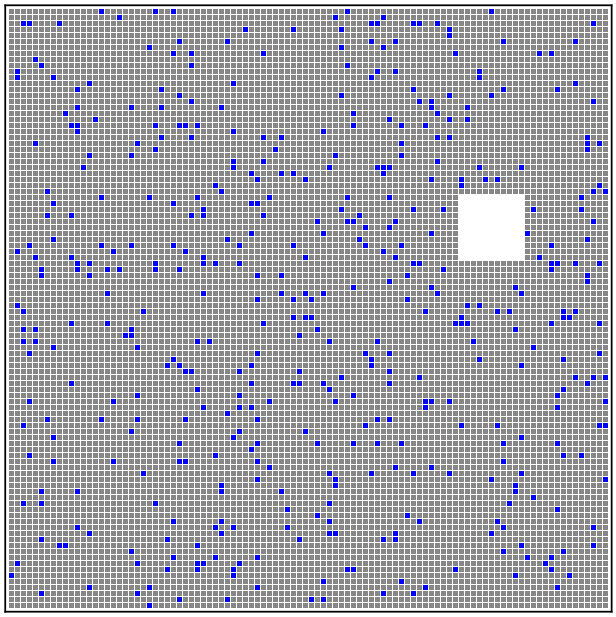


Figure 3.6: Snapshot of one replicate well-mixed population containing patches of cooperators (blue), cheaters (grey), and empty cells (white). Here, cooperators exist, but only as individuals surrounded by cheaters. Because of this, these small patches of cooperators are unable to produce enough public good to allow themselves to survive the kill event.

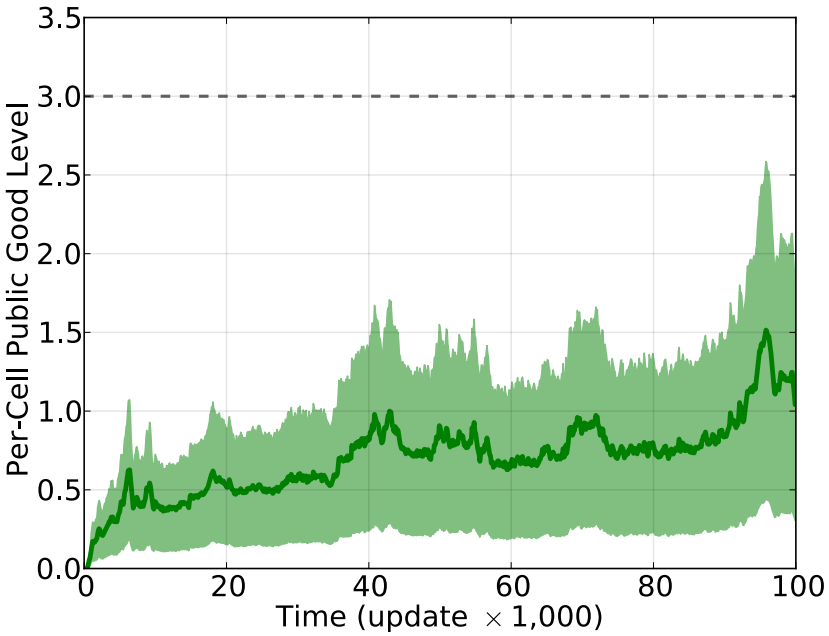



Figure 3.7: Average per-cell public good levels in well-mixed populations (n=49). Shaded region indicates $\pm 1$ standard error.

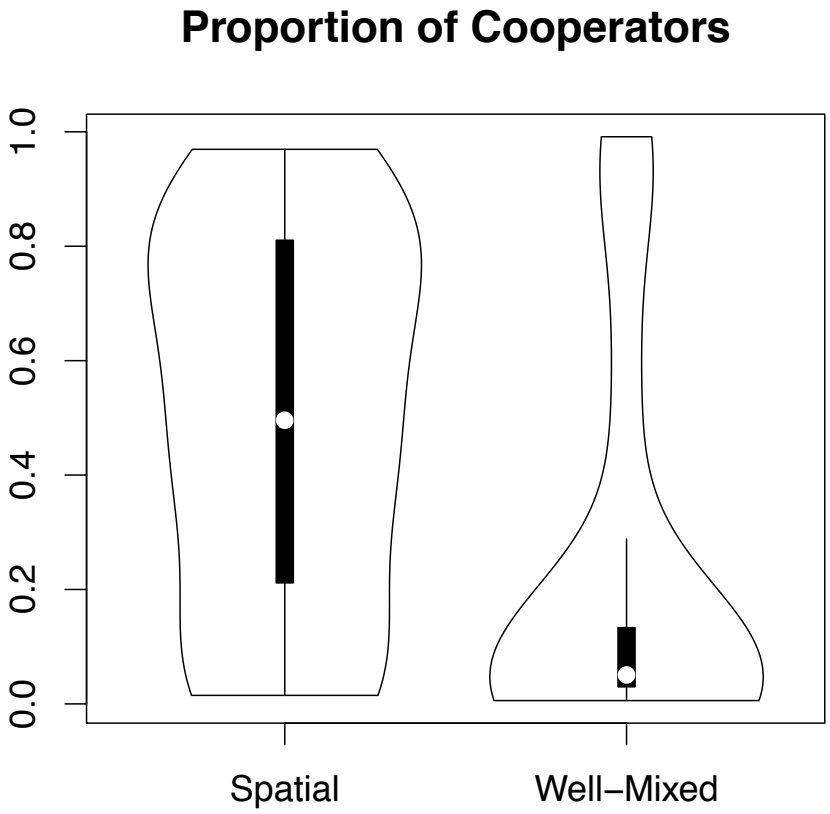



Figure 3.8: Proportions of cooperators in spatially-structured and well-mixed populations. In spatially-structured populations, 33 of 43 populations which persisted for 100,000 updates maintained at least 10% cooperators. Among well-mixed populations, only 12 of 48 maintained 10% cooperators.

by changing its radius, linear increases in radius led to geometric increases in the number of cells affected by the kill event. For example, a radius of 5 results in 121 cells being examined by the kill event, while a radius of 10 produces an area containing 441 cells, which is over 3.5 times larger.

All environments with kill radii above 5 cells proved to be too adverse in these experiments, however, and populations evolved in these environments were unable to persist. In these experiments, the kill event decimated populations before they were able to evolve a level of public good production necessary to survive. Here, even focusing solely could not compensate for the increased level of disturbance.

However, one way in which organisms could survive in these environments would be to migrate from other, more hospitable, environments, where they were able to evolve some level of cooperation. In these instances, organisms would simply need to adjust their cooperative effort to match the new environment instead of evolving it *de novo*.

As an approximation for populations migrating from one environment to a more adverse environment, populations were evolved in environments in which the radius of the kill event was incrementally expanded from 5 cells to 8, more than doubling the number of organisms at risk. The key parameters for these experiments are listed in Table 3.2.

Table 3.2: Virtual Biofilm Model Parameters for Increasing Disturbance Experiments

| Property | Value |
|---|---|
| Population Size | 10,000 |
| Replicate Populations | 50 |
| Updates | 100,000 (16,738.70 $\pm$ SEM 746.78 generations) |
| Kill Event Start Update | 1,000 updates |
| Kill Event Frequency | 1 update |
| Kill Event Radius | 5 (121 cells)–8 (289 cells) |
| Kill Event Focal Cells Per Update | 1 |
| Kill Event Public Good Threshold | 3 units |

These populations adapted to the changing environments and produced enough resource to survive the increasing disturbance. Figure 3.9a shows the composition of cooperators and cheaters in these populations. In response to the increasing disturbance in the environment, cooperators

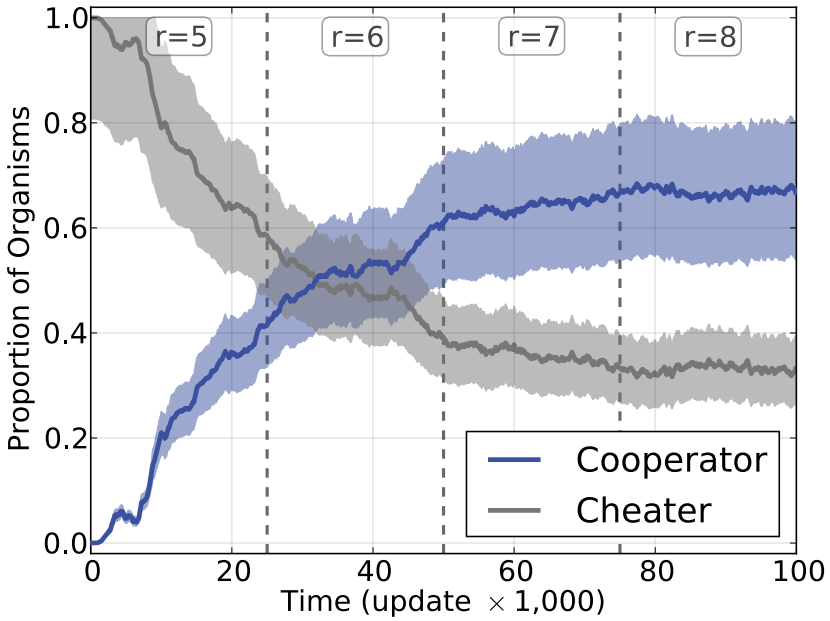
emerged as the dominant strategy at the end of these experiments, accounting for approximately 67% of the population. These cooperators produced an average level level of 10.61 units of public good per cell, enabling 98% of organisms, including cheaters, to avoid being killed, as shown in Figure 3.9b. For those populations that were driven to extinction by the kill event, Figure 3.10 shows the radius of the kill event corresponding to the time at which extinction occurred.

It is likely that these surpluses in public good played a key role in survival as adversity was increased through niche construction [94]. By creating surpluses in public good that spread throughout the environment, cooperators were able to alter their environment through niche construction, making it less adverse for future generations. This ensured that the number of their descendants that survive as disturbance increased was sufficient to avoid extinction. Such trans-generational cooperation has been shown by Lehmann [78] to reduce the effects of competition among kin in spatially-structured populations, and thereby aid in maintaining cooperation.

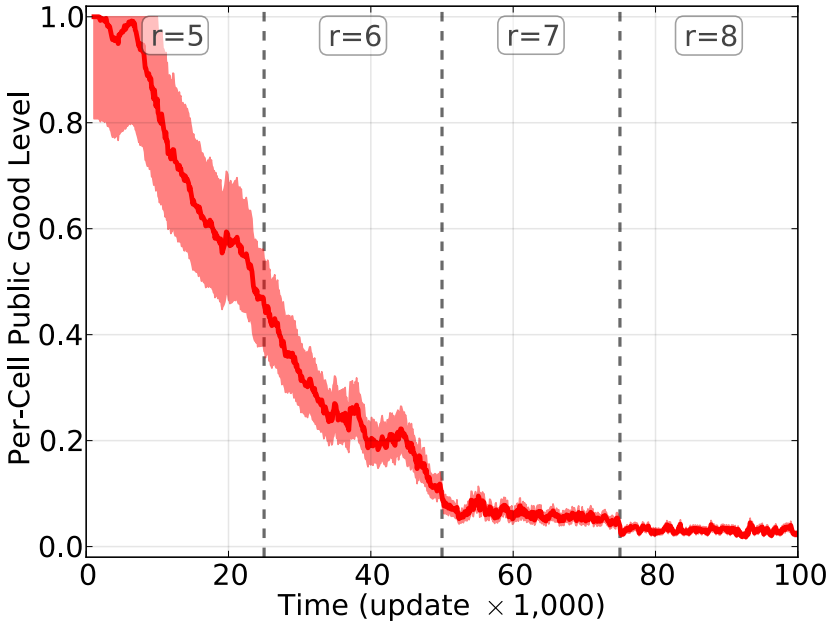
These results demonstrate that, through incremental changes in cooperative effort and niche construction, cooperation can be robust to increases in disturbance and more extreme environments. Decreases in disturbance poses different challenges to the evolution of cooperation, potentially to a point where the meager benefits provided by cooperation do not justify their costs. The next section examines whether cooperation can re-evolve in environments after cooperators have long been purged from environments where cooperation was not beneficial.

## 3.4 Temporary Loss of Disturbance

Since the 1950s, the problem of antibiotic resistance has grown to become a major public health concern. The widespread use of antibiotics has led to an ever-decreasing window of time between the introduction of an antibiotic and the development of resistance to that antibiotic [15]. Although perhaps impractical, one naïve potential strategy for tackling this problem would be to simply remove a particular class of antibiotics from use. Presumably, over time, infecting strains would purge the mechanisms related to resistance to that particular antibiotic from their genomes as they




(a)




(b)

Figure 3.9: Persistence of cooperators in increasingly-adverse environments. (a) Cooperators were able to evolve and account for approximately 60% of cells in the population (n=50) (b) In these populations, public good production allowed approximately 85% of organisms to avoid being killed by the kill event. By producing ample amounts of public good, cooperators were able to ensure the survival of populations as disturbance increased. Shaded regions indicate $\pm 1$ standard error.

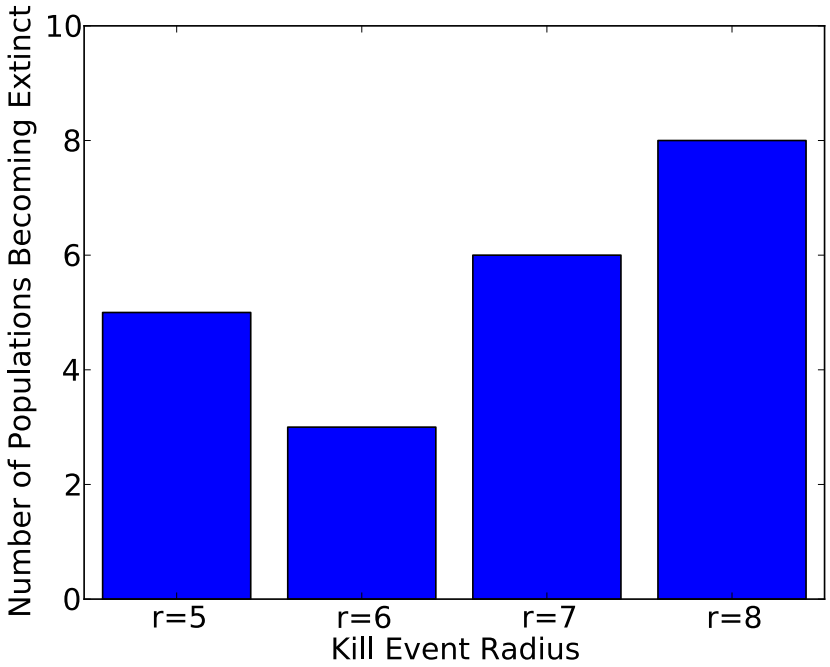



Figure 3.10: Number of populations driven to extinction for each level of disturbance when disturbance was periodically increased. Of the 50 populations studied, 26 survived for the duration of the experiment.

became non-beneficial. Over time, evolution could lead these strains further away to a point where, if that antibiotic were re-introduced, strains would not be able to re-adapt quickly enough to avoid being eradicated.

To examine how populations of cooperators would react to the absence and return of adversity, populations were evolved in an environment where the kill event was suspended during updates 40,000 through 60,000, or for roughly 3,500 generations. In these experiments, spatial structuring was maintained by placing offspring into neighboring cells, and the kill event used a 5-cell radius with a 3-unit threshold. The key parameters used for these experiments are listed in Table 3.3.

Figures 3.11a and 3.11b show the strategies maintained by populations and the fraction of organisms killed, respectively. During the respite, mean cellular resource levels of public good fell below the kill threshold through decay, leaving organisms vulnerable. Upon the return of the kill event, populations adapted, and cooperative public good production re-evolved in 2 of the 46 populations that experienced this respite. In this Avida-based model, it is likely that cooperation was able to quickly re-emerge through the fixation of a small number of mutations in lineages

Table 3.3: Virtual Biofilm Model Parameters for Loss-of-Disturbance Experiments

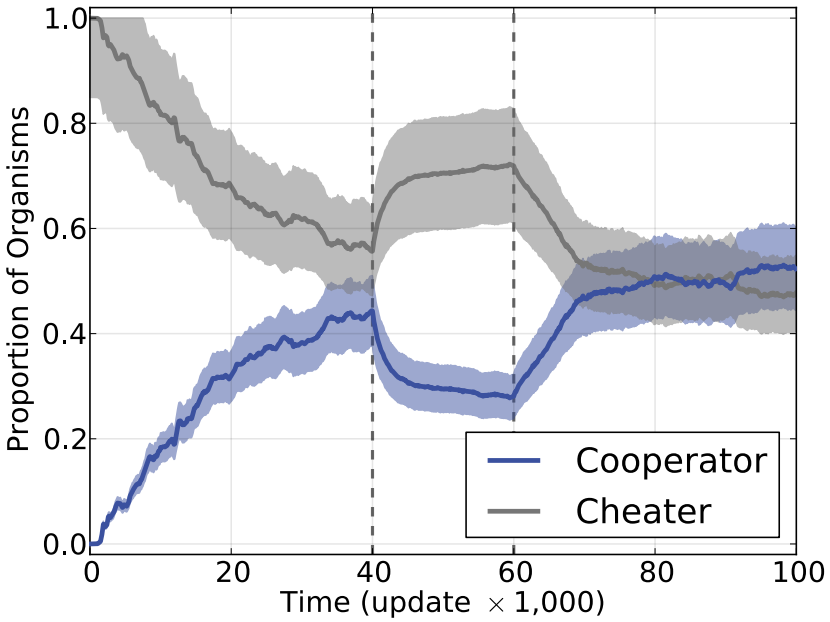
| Property | Value |
|---|---|
| Population Size | 10,000 |
| Replicate Populations | 50 |
| Updates | 100,000 (17,484.25 ± SEM 517.27 generations) |
| Kill Event Start Update | 1,000 updates |
| Kill Event Frequency | 1 update |
| Kill Event Inactivity | Updates 40,000–60,000 |
| Kill Event Radius | 5 |
| Kill Event Focal Cells | 1 |
| Kill Event Public Good Threshold | 3 units |

performing a similar logic task that allowed the cooperative task to be completed.

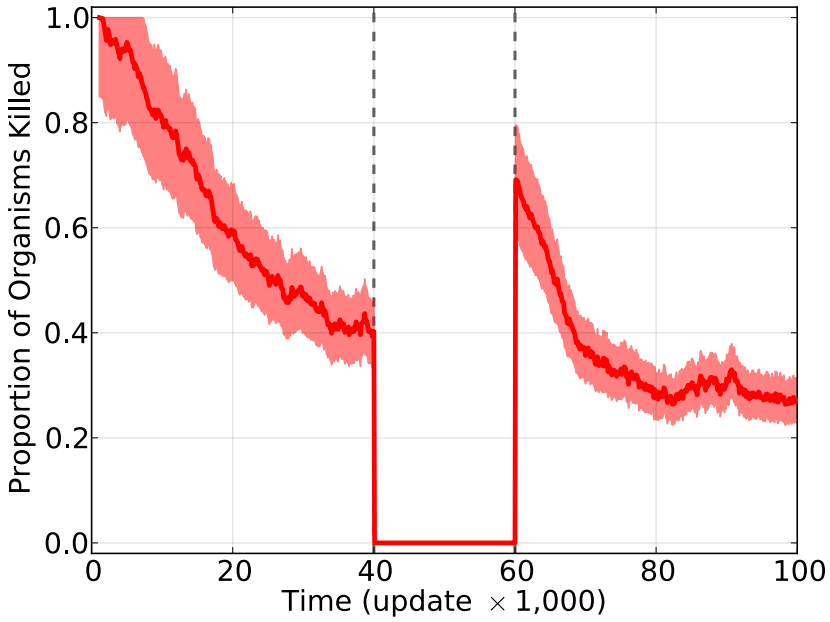
This model demonstrates that when selection for those traits resumes, populations can quickly re-evolve cooperative, resistant behaviors that have been long absent in order to occupy that niche, which could have significant implications for future treatments of bacterial infections. For instance, even when an "anti-infective" treatment of interrupting the behaviors of bacteria is used instead of killing the bacteria, selection may still act strongly to select for resistance to these treatments if it allows these resistant strains to occupy a niche [33]. The specific response to these types of treatments is not currently known; however, models like the one presented here may offer important insights into how selection operates under such treatments.

## 3.5 Population Bottlenecks

Population bottlenecks are a form of disturbance in which the size of a population is drastically reduced. In nature, bottlenecks are created by both environmental change and interactions among species, and may occur over both short- and long timescales. As an example, the Hawaiian Bobtail Squid (*Euprymna scolopes*) lives in a symbiotic relationship with the marine bacterium *Vibrio fischeri*, whose bioluminescence allows the squid to avoid detection through counterillumination [124]. Each morning, the *V. fischeri* residing in the squid's light-emitting organ experience a bottleneck, whereby over 95% of the population is expelled into the sea [77]. This reduces the




(a)




(b)

Figure 3.11: The evolution and maintenance of cooperation when the kill event was suspended between updates 40,000 and 60,000 (n=44): (a) During this respite, cheaters gained in abundance at the expense of cooperators, which were no longer favored. (b) Although the number of organisms killed increased after disturbance was re-introduced, the increase in cooperators quickly allowed populations to persist through the production of public good in all but two populations. Shaded regions indicate $\pm 1$ standard error.

cost of the symbiotic relationship for the squid, which does not benefit from bioluminescence during the day. This bottleneck may also enable the population of *V. fischeri* to stave off invasion by cheaters, which would compete for the benefits provided by the symbiotic relationship.

Frequent bottlenecks, such as those experienced daily by populations of *V. fischeri*, have been shown to promote the evolution of cooperation. By propagating different volumes from cultures of biofilm-forming *Pseudomonas fluorescens* at fixed intervals, Brockhurst demonstrated that smaller volumes, and therefore larger bottlenecks, produced greater proportions of biofilm-forming cooperators [6]. These experiments concluded that bottlenecks had benefited cooperators by increasing relatedness among passaged cells, particularly cooperators. It should be noted that in these populations of *Pseudomonas fluorescens*, the biofilm-forming cooperator phenotype is ecologically dominant, so it was able to grow to reach greater densities than cheaters, which only begin to gain in abundance once they can exploit the biofilm produced by cooperators. Due to the bottleneck imposed during propagation, the number of cheater cells was reduced, which, in turn, lessened competition, and allowed cooperators to flourish.

Similarly, Griffin et al. [49] demonstrated that when populations of *Pseudomonas aeruginosa* were passaged through a global competition phase containing cells drawn from all subpopulations, cooperators were more abundant in the resulting populations than when was done independently within each subpopulation [49]. In their experiments, which were designed to study how the scale of competition [44] affects the evolution of cooperation, bottlenecks that included global competition allowed cooperative siderophore production to be favored. This is because the competition phase required cooperation in order to better compete during this phase. When competition was localized to each replicate population, lower levels of cooperation resulted, because cooperation no longer significantly increased the probability of propagation.

Both of the studies described above used well-mixed populations. To determine the effects that population bottlenecks have on the evolution of cooperation in spatial populations, experiments were conducted using a computational model in which populations were periodically subjected to bottleneck events. Contrary to previous findings, the results of experiments using this model

demonstrate that bottlenecks can either help or hinder the evolution of cooperation, depending on the spatial distribution of cells and a number of other factors.

### 3.5.1 Spatial Population Growth and Bottleneck Model

For these experiments, a model similar to the virtual biofilm model presented in Section 3.1 was constructed using the SEEDS platform. In this cellular-automaton model, each cell in the population was either *empty* or contained either a *cooperator* or a *cheater*. These three cell types are interchangeably referred to as *strategies* or *phenotypes*. Upon initialization, approximately 1% of cells were randomly assigned to be either a cheater or cooperator, and the remaining cells were made empty. This yielded approximately 9,900 empty cells, 50 cooperator cells, and 50 cheater cells for each 10,000-cell bounded lattice population.

During each time step, the cells in the population were updated asynchronously. When a cheater cell was updated, it reproduced with probability 0.1. Reproduction is discussed further below. When a cooperator cell was updated, it reproduced with probability 0.095. This 5% decrease in fecundity was attributed to the cost of producing a public good. When updated, cooperators produced 0.9 units of this public good, which was placed into the environment at that cooperator cell's location. This public good decayed at a rate of 0.01 and diffused at a rate of 0.01.

During each time step, a kill event selected a cell at random from the environment, examined the level of public good in that cell and all cells within a 5-cell radius, and killed any cooperator or cheater residing in those cells if the average level of public good was below 3 units. When a cell was killed, it became empty.

Space served as a strict limiting resource. Unlike many other computational models, reproduction could only occur into neighboring empty cells. If no neighboring cells were available during reproduction, no offspring was produced. Otherwise, the neighboring empty cell adopted the phenotype of its parent. To allow for continuous growth and evolution, populations were diluted to 1% every 200 time steps. During this bottleneck process, each cell was killed indiscriminately with probability 0.99. The public good was also diluted by reducing its level by 99% uniformly over

space. The period of time between dilutions is referred to below as a *growth cycle*. Fifty replicate populations were evolved in this environment for 10,000 time steps. Table 3.4 lists the properties and their values used in these experiments.

Table 3.4: Growth and Bottleneck Model Parameters

| Property | Value |
|---|---|
| Population Size | 10,000 |
| Replicate Populations | 50 |
| Time Steps | 10,000 |
| Cheater Fecundity | 0.1 |
| Cooperator Fecundity | 0.095 |
| Population Dilution | 1% |
| Resource Dilution | 1% |
| Public Good Contribution | 0.9 units per time step |
| Public Good Decay Rate | 1% per time step |
| Public Good Diffusion Rate | 1% per time step |
| Growth Cycle Duration | 200 time steps |

Because cells could not overwrite other cells, this model offered a clear picture of growth, which enabled the fitnesses of cooperators and cheaters to be estimated. After Lenski et al. [79], the average rate of increase, or Malthusian parameter $m$, for cooperators and cheaters during one 200-time-step growth cycle is shown in Equation 3.1, where $N_S(f)$ represents the cell count of strategy $s$ at time step $f$, the end of the growth cycle, and $N_S(i)$ represents the cell count of strategy $s$ at time step $i$, the beginning of the growth cycle.

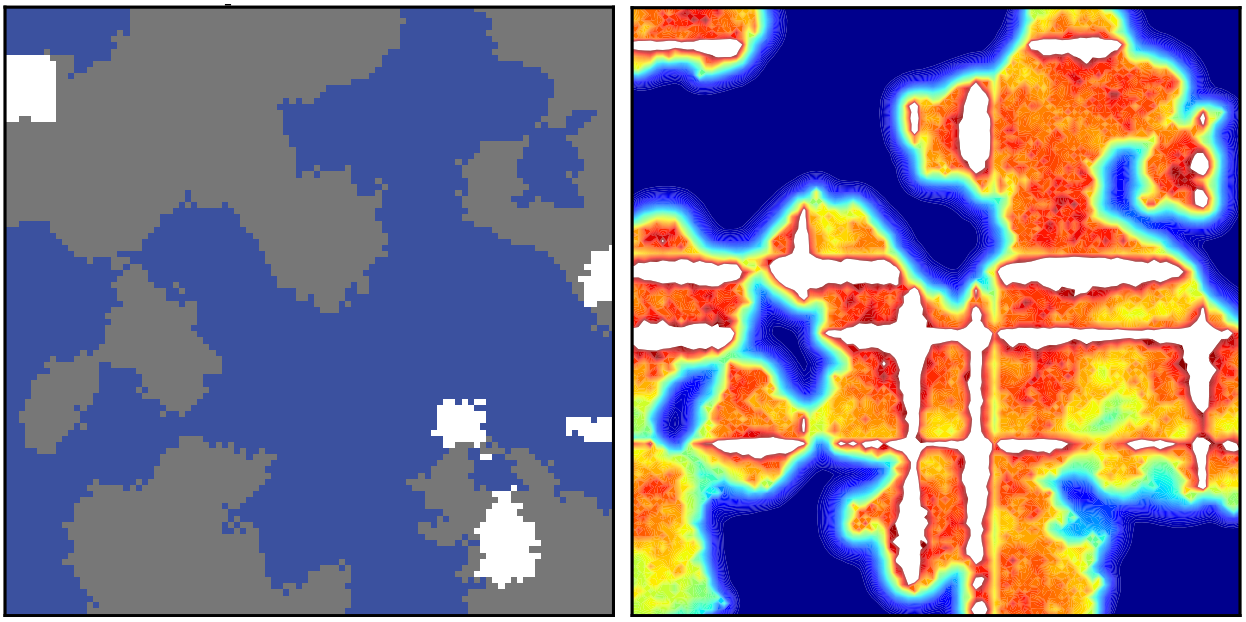
$$m_S = ln[N_S(f)/N_S(i)]/200 \tag{3.1}$$

The fitness $W$ of the cooperator phenotype relative to the cheater phenotype during each growth cycle was estimated by comparing these rates of increase, as shown in Equation 3.2, where $c$ and $\bar{c}$ represent the cooperator and cheater strategies, respectively. Values above 1 indicated that cooperators were favored, while values below 1 indicated that cheaters were favored. It is important to note that because of the spatial structuring of populations, these fitness estimates reflect the fitness of cooperation averaged over space.

$$W_i = m_C/m_{\bar{C}} = \frac{ln[N_C(f)/N_C(i)]}{ln[N_{\bar{C}}(f)/N_{\bar{C}}(i)]} \tag{3.2}$$

### 3.5.2 Growth and Cooperation in Response to Bottlenecks

The cooperative phenotype was maintained in 19 of the 50 replicate populations evolved in this environment. As demonstrated in Figure 3.12, all populations showed a great deal of positive assortment due to the spatial growth process, forming large clusters of cooperators and cheaters. These large clusters benefited cooperators in two ways. First, cooperator cells within these clusters produced large amounts of the public good, as shown in Figure 3.12b, which allowed them to survive the kill event. Second, because cells passing through the bottleneck were not perturbed, a large distance separated cooperator cells that had been located the interior of these large clusters from cheaters. This separation allowed patches of cooperators to grow without competition from cheaters at the beginning of each cycle.



(a) (b)

Figure 3.12: State of the population and public good prior to dilution in one example population. (a) Spatial distribution of cooperator (blue), cheater (grey), and empty (white) cells. (b) Distribution of public good in the environment. High levels of public good can be seen near patches of cooperators, while low levels of public good exposed cheaters to the kill event.

As depicted in Figure 3.13, growth in this model resembled that of bacterial populations. At the beginning of each growth cycle, populations grew exponentially, filling the available space in the environment. Once this space resource depleted, growth slowed, and a stationary phase began, where populations maintained that density until the subsequent bottleneck was introduced. Because death was not captured in this model, aside from the kill event, these populations did not experience a death phase.

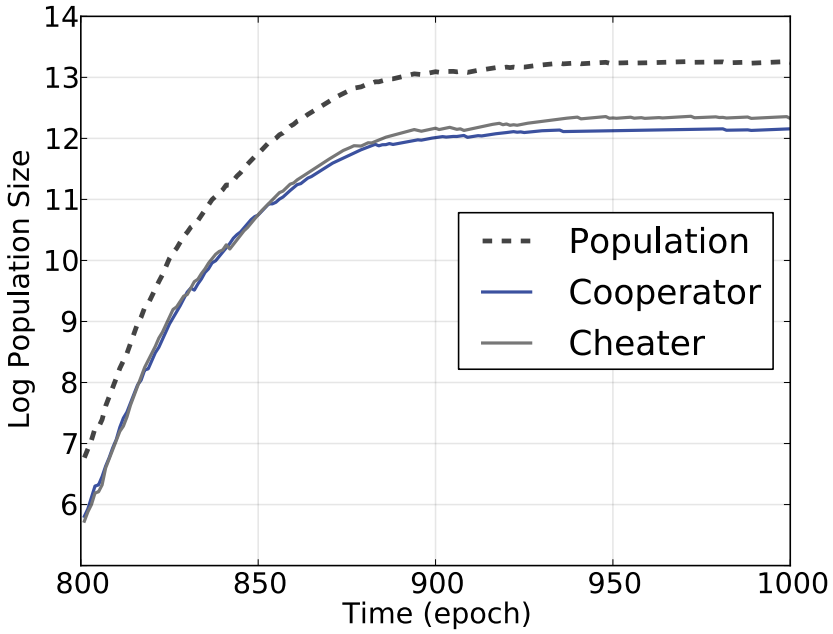
As shown in Figure 3.14, populations reached significantly greater densities when cooperators persisted (9,120.26 $\pm$ SEM 95.98) than when only cheaters survived (8,491.58 $\pm$ SEM 12.75) ($p < 0.0001$; Welch Two Sample t-test). This result is common among cooperative behaviors, as cheaters generally follow a high-growth strategy, while cooperation allows for high yield (e.g. [103]).

These experiments demonstrated that cooperation can be maintained and allow populations to reach significantly larger sizes in this environment. However, a greater understanding of the difference in fitnesses between the cooperator and cheater phenotypes in spatially-structured populations is required to understand why 19 replicate populations maintained cooperation and 31 did not—even though these experiments offered the same amount of space, started from approximately the same numbers of cooperators and cheaters, and modeled cooperation using the same costs. The next section addresses these difference in fitness over time and space.

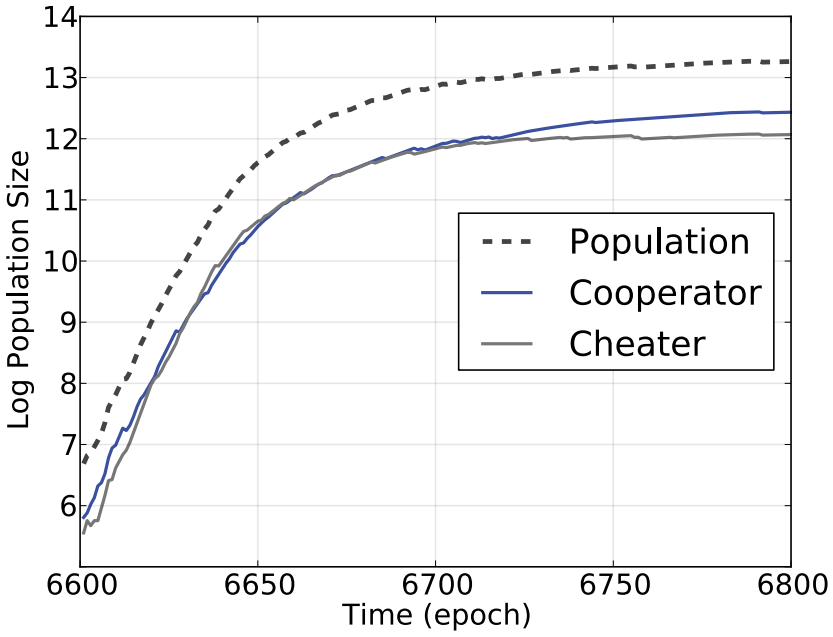
### 3.5.3 The Ever-Changing Targets of Selection

The fitness of both strategies depended on a myriad of factors in these experiments. Among them were the differences in growth rate between cooperators and cheaters, the amount of public good present in an area, the composition of neighboring cells, the probability of being selected by the kill event, and the availability of empty neighboring cells. Because many of these factors changed continually, so did fitness. These factors also varied across space, so selection could favor cooperation in one region of the environment while strongly opposing it in another.

The relative fitness of cooperation was calculated for each growth cycle following Equation




(a)




(b)

Figure 3.13: Growth of cooperators (blue), cheaters (grey), and the population as a whole (dashed) during one growth cycle for one representative population: (a) Although cooperators had a greater initial abundance, the faster growth rates of cheaters allowed them to reach a larger density. (b) Faster growth rates did not always lead to greater abundances. Here, the growth of cheaters was limited by the kill event, which occasionally allowed cooperators to reach greater abundances.

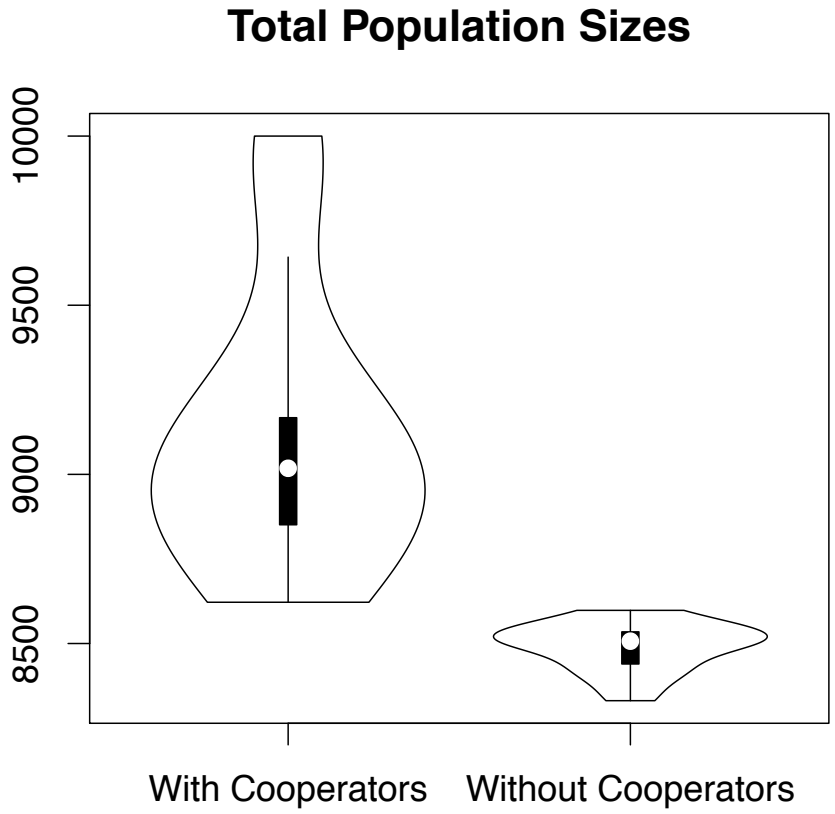



Figure 3.14: Final population sizes in populations that maintained cooperation (n=19) and those that did not (n=31). Through the production of public good, which enabled cells to avoid the kill event, populations achieved significantly larger densities when cooperation was maintained.

3.2. In terms of bottlenecking, when the relative fitness of cooperators was greater than 1, the proportion of cooperators able to pass through the population bottleneck was expected to increase. As discussed in the next section, the proportion of cooperators that pass through the bottleneck is a strong determinant of whether or not cooperation persists.

As shown in Figure 3.15, the fitness of cooperators relative to cheaters varied throughout the duration of an experiment. In most populations, selection cycled between favoring cooperators and cheaters. When cooperators were favored, they produced an abundance of public good, which cheaters subsequently exploited. However, as cheaters increased in abundance, they became more susceptible to the kill event, which freed space in which cooperators could grow. Because of the spatial variations that existed among populations, the relative fitness of cooperation followed a unique trajectory for each population. Because of this, cooperation was able to persist in some populations, while being driven to extinction in others. Figure 3.16 shows the changes in abundance of cooperators and cheaters in two populations throughout an experiment, demonstrating each of these outcomes.

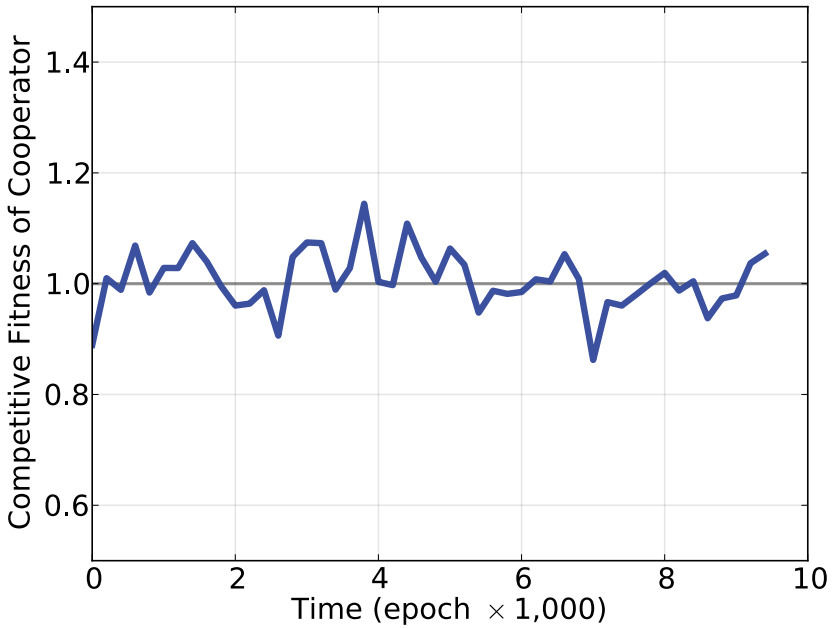
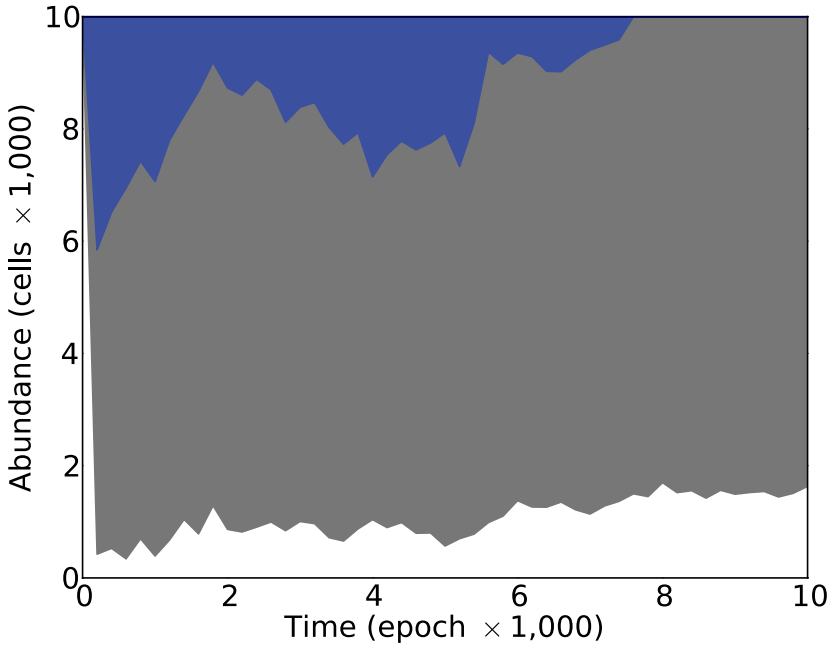
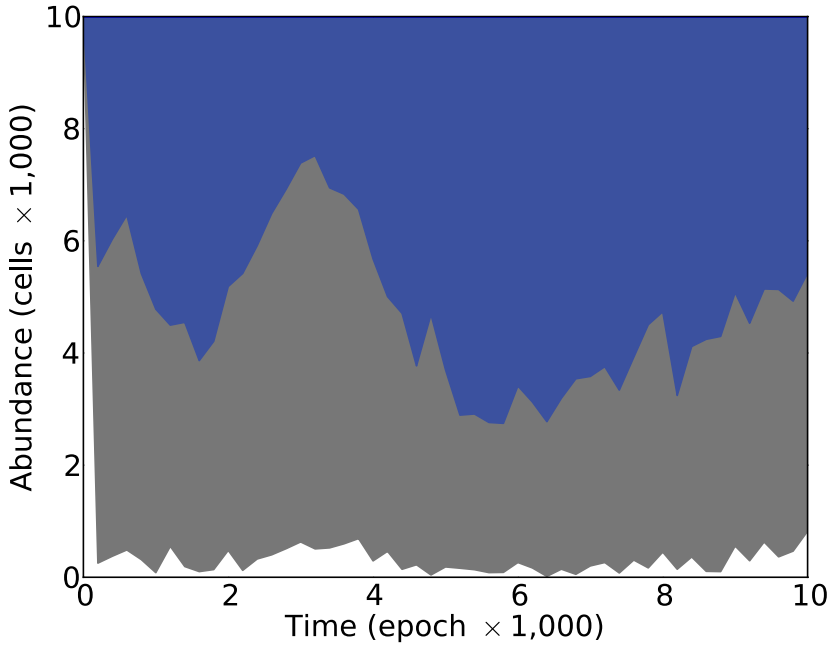



Figure 3.15: Relative fitness of cooperation over time in one replicate population in which cooperation was maintained. Values greater than 1 indicate conditions when cooperation was favored.

Figure 3.15 is a clear demonstration of what motivates this dissertation: By continually af-




(a)




(b)

Figure 3.16: Abundances of cooperators (blue), cheaters (grey), and empty cells (white) over the duration of an experiment. (a) In this population, the benefits of cooperation are not able to compensate for the costs, and cooperators are eventually excluded. (a) Due to different distributions of cells in space, this population was able to maintain cooperators and grow to larger densities.

fecting the benefits that cooperative behaviors provide, as well as the associated costs, ecological factors, such as those presented here and in the following chapters, can have a significant impact on the evolution of cooperation, ultimately determining whether or not cooperation can be maintained in that environment. Because of this, models of cooperation that represent the fitness effects of cooperation, such as its costs and benefits, as fixed values cannot accurately capture the evolutionary dynamics of cooperative behaviors. It is only by understanding how ecological factors affect cooperation can we effectively begin to disentangle the origins, the evolution, and the maintenance of cooperation.

Figure 3.15 also demonstrates that there are many instances when *not cooperating* is beneficial. In systems where this occurs, *facultative* cooperators, who choose to cooperate only when it is likely to be beneficial, would have a selective advantage over *obligate* cooperators, such as those modeled in these experiments. Previous studies have shown that even simple mechanisms, such as expressing cooperative behaviors stochastically through phenotypic noise, can allow cooperation to be maintained [1]. Even sources of novelty, such as mutation, migration and drift, the effects of which are examined in Section 5.7, can help to maintain cooperation in these environments. Several more complex mechanisms that enable facultative cooperation have been identified in microorganisms [118]. One such mechanism is *quorum sensing*, which enables cooperators to cooperate only when their density is great enough to provide sufficient benefit [28, 35, 57]. Alternately, microorganisms display a variety of mechanisms for preferentially cooperating with close relatives, such as greenbeards [109], poison-antidote systems (e.g. [12] and see Chapter 5), and positive assortment [100].

This section discussed the complex variations of fitness occurring in spatial populations. The use of bottlenecks in this growth model enabled the relative fitness of cooperation to be estimated at the population level, demonstrating how cooperation can be both favored and selected against when ecological factors are at play. However, the effect that the bottleneck itself had on the evolution of cooperation was not examined. Previous investigations using well-mixed populations have shown that bottlenecks promote the evolution of cooperation either by reducing competition from cheaters

[6] or maintaining positive selection for the cooperative benefit [49]. As the previous two sections have demonstrated, the variations in conditions within populations that spatial structuring produces can create dynamics that differ significantly from well-mixed populations. The following section examines the role that population bottlenecks play in the evolution of cooperation in spatially-structured populations. The results of these experiments provide first insights into a process which may be equally- or even more important in maintaining cooperation in spatial populations.

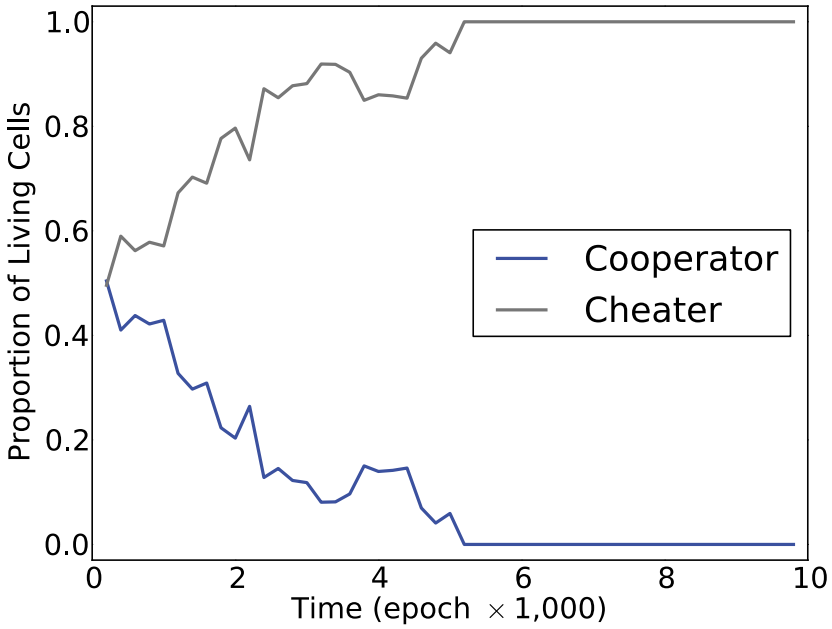
### 3.5.4 Bottlenecks and Sampling

Because the organisms chosen to pass through the bottleneck are selected indiscriminately, the bottleneck is essentially a random sampling process [58]. Therefore, in order to maintain presence in this environment, it is imperative that each strategy reach a population density large enough to be well represented after the bottleneck.

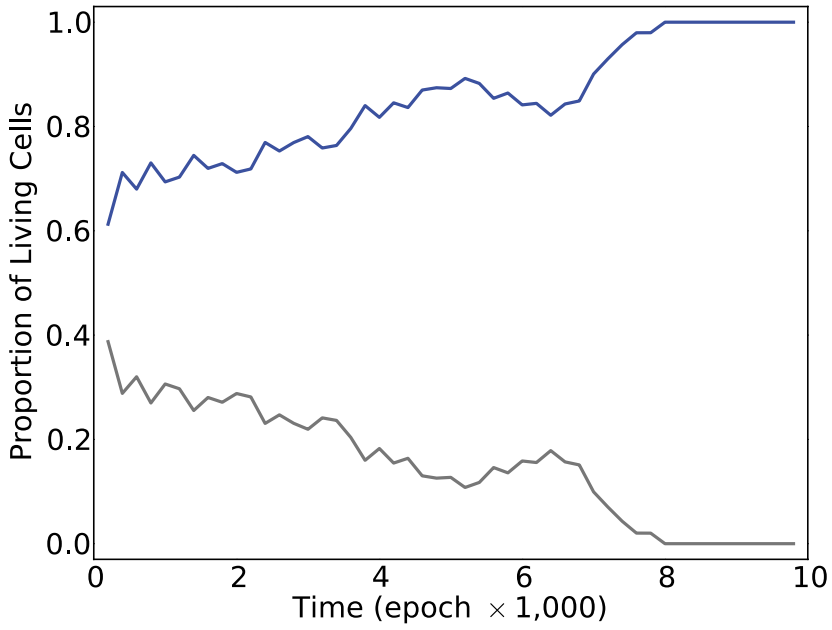
In this iterated sampling process,even if growth rates were identical, a strategy that is underrepresented at the beginning of a growth cycle is likely to make up an even smaller proportion of the population at the end of the cycle, due to exponential growth. As this process proceeds, its effect becomes more pronounced, and, after several cycles, that underrepresented strategy may reach such low abundances that it does not pass through the bottleneck, at which point it becomes extinct. Low abundances can also create founder effects, where the resulting populations are not representative of the populations from which they came, [84].

The effects of this process are shown in Figure 3.17, where the proportion of the dominant strategy increases continually, while the underrepresented strategy is driven to extinction. Although bottlenecks more often allowed cheaters to exclude cooperators, this process enabled cooperators to exclude cheaters in one of the fifty replicate populations, as shown in Figure 3.17b.

This problem of sampling becomes even more pronounced when the underrepresented strategy also has a slower growth rate, which is the case for cooperators in this model. For these strategies, some method is required that enables that strategy to remain present in the subset of cells chosen to carry on to the next cycle. Figure 3.18 shows that in one population, the slower-growing cooperator




(a)




(b)

Figure 3.17: Proportions of cooperators (blue) and cheaters (grey) among the cells passing through the bottleneck over time. Often, whichever strategy became dominant first maintained that dominance for the duration of the experiment, whether cheaters (a) or cooperators (b).

cells increased in proportion when public good production enabled them to persist while the kill event simultaneously removed a significant portion of cheaters.

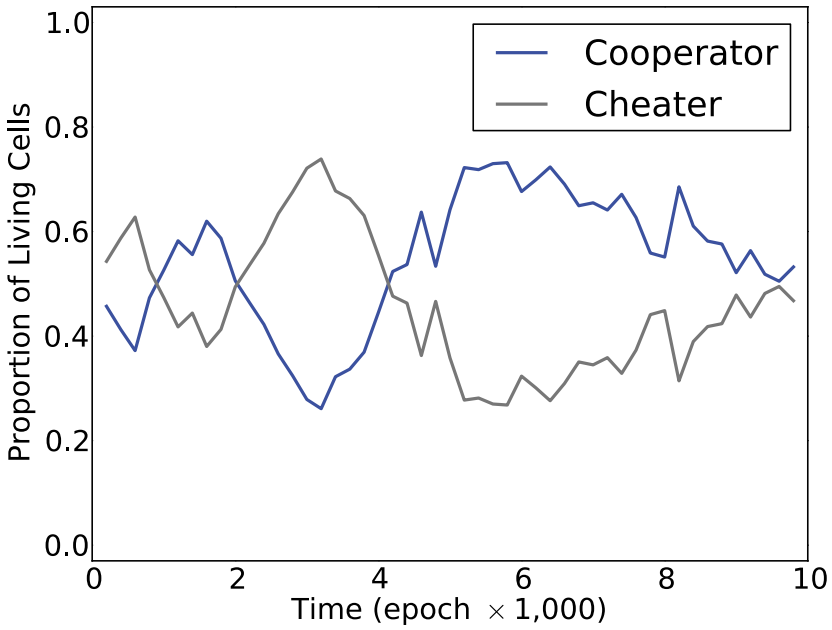



Figure 3.18: Proportions of cooperators (blue) and cheaters (grey) among the cells passing through the bottleneck over time in one population, where the benefits of cooperation allowed cooperators to recover after several invasions by cheaters.

These results demonstrate that indiscriminate population bottlenecks act as a non-biased random sampling process, which can either promote cooperation or work to quickly purge it from the population. The outcomes of this bottlenecking process, therefore, depend entirely on the ability of the cooperative phenotype, or any phenotype for that matter, to maintain a competitive abundance in the population, which, in turn, depends on each of the components that affect fitness, as discussed in Section 3.5.3. Because of the exponential growth phase that occurs after sampling, population bottlenecks simply act to magnify the differences in abundance between strategies.

This conclusion is more general than that proposed in a previous study, which demonstrated that larger bottlenecks consistently produced populations with greater proportions of cooperators [6]. This difference is likely due to differences in model systems. In this previous work, cooperators were ecologically dominant, and cheaters only began to have an effect once there was a substantial biofilm that could be exploited. Because cooperators represented the majority in this system,

the bottleneck benefited them by halting the invasion by cheaters before they could reach larger densities.

In contrast, cheaters and cooperators in the model presented here began on even footing. In combination with their faster rate of growth, this made it much easier for cheaters to grow to larger densities than cooperators. When this occurred, cooperators were usually driven to extinction. However, when the production of good enabled cooperators to maintain adequate population sizes, while cheaters were constrained by the kill event, cooperators could persist and even exclude cheaters, as was shown in Figures 3.18 and 3.17b, respectively.

From this understanding, it is clear that the frequency at which the bottleneck occurred can have a great impact on outcomes. As shown in Figure 3.13, populations typically reached stationary phase during each growth cycle after approximately 100 time steps. Cooperators benefited from a bottleneck frequency of 200 time steps, therefore, when the kill event decreased the density of cheater cells during this stationary phase. Cooperators further benefited if the kill event affected adjacent patches of cheater cells, which allowed cooperators to occupy some of the resulting space. The next section addresses how the frequency of bottlenecks affects the ability of the cooperator phenotype to reach competitive densities and ultimately persist in environments with different bottleneck frequencies.

### 3.5.5 Bottleneck Frequency

To observe the affect of bottleneck frequency on the maintenance of the cooperator phenotype, experiments were conducted using bottleneck frequencies ranging from 50 time steps to 1,000. The hypothesis was that when dilutions were more frequent, not allowing cooperators to benefit as much during stationary phase, cooperation would persist in fewer populations. However, when stationary phase was extended, the benefits of cooperation would become more pronounced, and cooperators would be maintained in more populations. For each bottleneck frequency, 50 replicate populations were evolved.

As shown in Figure 3.19, the results of these experiments support the hypothesis that less

frequent bottlenecks allow cooperators to persist in these environments. When the bottleneck occurred every 50 time steps, populations were not able to reach sufficient densities to survive and maintain either strategy.

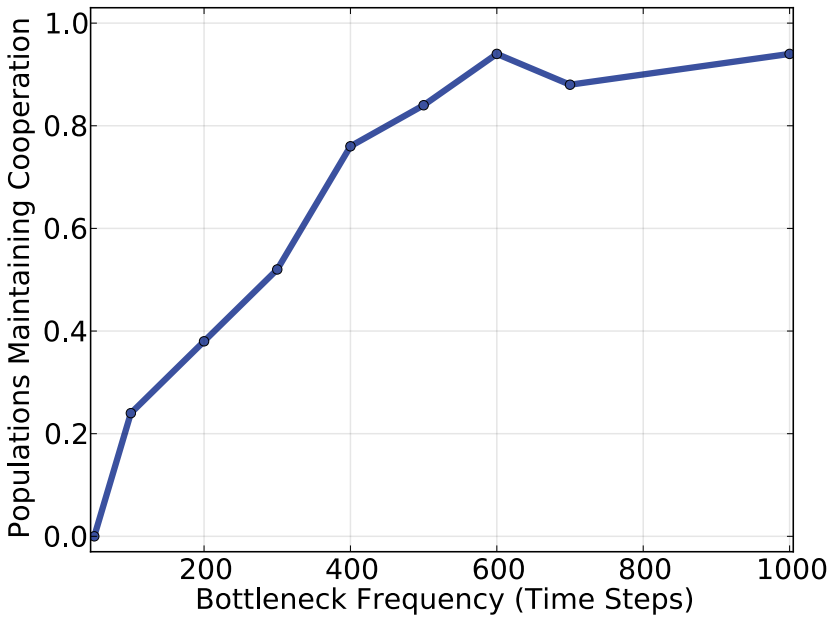



Figure 3.19: Proportion of populations maintaining cooperation with different bottleneck frequencies

In control experiments where dilution was amortized, killing each cell with probability $0.01/200 = 0.00005$ per time step (data not shown), populations were also not able to survive. This further demonstrates the importance of growth that occurs between population bottlenecks.

## 3.6 Discussion

Environmental disturbance effects all forms of life in one way or another. This section has presented a number of experiments that provided new insights into how disturbance affects the evolution of cooperation using a model inspired by bacterial biofilms.

First, the results presented in Section 3.2 showed that the spatial structuring of organisms can significantly affect the evolution of cooperation. Because of this, spatially-structured populations were used for all of the experiments presented in this dissertation.

Section 3.3 examined how populations responded to increasing levels of disturbance. The results of these experiments showed that populations were not able to survive when evolved in very diverse environments. However, positive niche construction through the production of surplus public good allowed populations to persist in environments where disturbance was incrementally increased.

When disturbance was lost, Section 3.4 showed that cheaters quickly eliminated cooperators from the population, as cooperation was no longer beneficial. However, once disturbance resumed, cooperators were able to quickly evolve and stay in the population, even after 3,500 generations. These results demonstrate that if the environment provides a niche that cooperators can occupy, cooperation can exist stably.

Finally, Section 3.5 examined how bottlenecks, disturbance events in which a significant portion of the population is killed, affect the evolution of cooperation in spatial populations. Although a previous study had shown that bottlenecks act to promote cooperation by reducing competition from cheaters [6], results from a growth-based model with periodic bottlenecks instead showed a more general effect of bottlenecks—that bottlenecks simply act to amplify the differences in abundance between two or more strategies. Whether or not bottlenecks aid cooperation depends solely on the ability of the cooperator strain to maintain abundances equal to or greater in size than cheaters. Although, bottlenecks were demonstrated to promote cooperation when they occurred at a rate that allowed cooperators to reach larger densities due to the benefits provided by public good production.

# Chapter 4

# Resource Abundance

The availability of a required resource plays a considerable role in determining evolutionary outcomes [121]. Presumably, as resource becomes more abundant, the relative costs of its allocation towards cooperation decrease, allowing organisms to perform cooperative acts in addition to their self-preserving and competitive activities. When that resource becomes more scarce, however, organisms may be forced to focus solely on their own interests. Previous experiments with microbial populations have shown this relationship [8], as well as a link between resource availability and diversification [52].

The work presented in this chapter focuses on this relationship between resource abundance and cooperation through the production of a public good. Here, prior microbial studies [8] motivated further investigation using digital models [20], which in turn prompted additional microbial experiments.

Section 4.1 details studies examining the relationship between resource abundance and cooperation using the virtual biofilm model. The results of these studies, which differed from previous studies, prompted further microbial studies, which are described in Sections 4.2 and 4.3.

## 4.1 Virtual Biofilm Experiments

### 4.1.1 Resource Availability in Virtual Biofilms

To examine the role that the abundance of a required resource plays on the evolution of cooperation, the virtual biofilm model introduced in Section 3.1 was used [19]. Although the fixed amount of space always serves as an important limiting resource in Avida, an additional required resource was defined. Unlike the experiments described in Chapter 3, the completion of both the cooperative and rewarded tasks required a sufficient amount of this resource in order to be executed. Completion of each task consumed 1 unit of this resource. Populations were evolved in environments with different levels of resource, which continually flowed into each cell in the environment at rates that yielded predetermined equilibrium levels. Resource was lost through a 1% diffusion rate into neighboring cells and through decay, which also occurred at a rate of 1% per update. Individuals were unable to directly sense the level of resource in their environment or the presence of neighboring individuals.

As in Chapter 3, the kill event began at update 1,000 and examined the mean level of public good in all cells within a distance of 5 cells from a randomly-chosen focal cell. If the average level of public good among these cells was below 3, all organisms residing in that region were killed. Organisms in regions with an above-threshold level were spared, regardless of whether they were cooperators.

In each of 27 different resource environments, which ranged from equilibrium levels of 2 units of resource per cell to 90, thirty replicate populations consisting of up to 10,000 organisms were evolved for 100,000 updates, or an average of 17,600 generations, starting from a single ancestor. Each population started with a different seed for the pseudorandom number generator, which altered the outcomes of stochastic processes and allowed each population to follow different trajectories. The key parameters for these experiments are listed in Table 4.1.

The kill event drove a portion of these populations to extinction in all resource environments. This occurred early in the experiments, when these populations had not grown to sufficient den-

Table 4.1: Virtual Biofilm Model Parameters for Resource-Based Experiments

| Property | Value |
|---|---|
| Population Size | 10,000 |
| Replicate Populations | 20 |
| Updates | 100,000 (17,602.51 ± SEM 129.93 generations) |
| Kill Event Start Update | 1,000 updates |
| Kill Event Frequency | 1 update |
| Kill Event Radius | 5 (spatial), 0 (non-spatial) |
| Kill Event Focal Cells Per Update | 1 |
| Kill Event Public Good Threshold | 3 units |
| Public Good Diffusion Rate | 0.01 |
| Public Good Decay Rate | 0.01 |
| Copy Mutation Probability | 0.0075 |
| Insertion Mutation Probability | 0.05 |
| Deletion Mutation Probability | 0.05 |
| Equilibrium Resource Levels (per cell) | 2–90 units |
| Resource Consumed per Task | 1 unit |

sities or had not yet evolved the production of public good. No relationship was found between resource abundance and the proportion of these populations that became extinct. Data for these populations are not included in the following results.

### 4.1.2 Results

As is shown in Figure 4.1, these digital populations showed a rapid transition from non-cooperation to cooperation at a critical level of resource. In this system, the transition occurred in environments with equilibrium levels of approximately 20 units per cell. Those populations evolved in resource-poor environments showed a uniformly low amount of cooperation, while those evolved in resource-rich environments cooperated at a uniform level that was significantly higher. Neither of these groups, however, was significantly different than populations at the intermediate level of 20 units per cell.

Although these data support the results of a previous by Brockhurst, et al. [8], the interpretations that they offer differ significantly. This previous study concluded that cooperation increased steadily as resource became more abundant [8]. From the perspective of inclusive fitness and

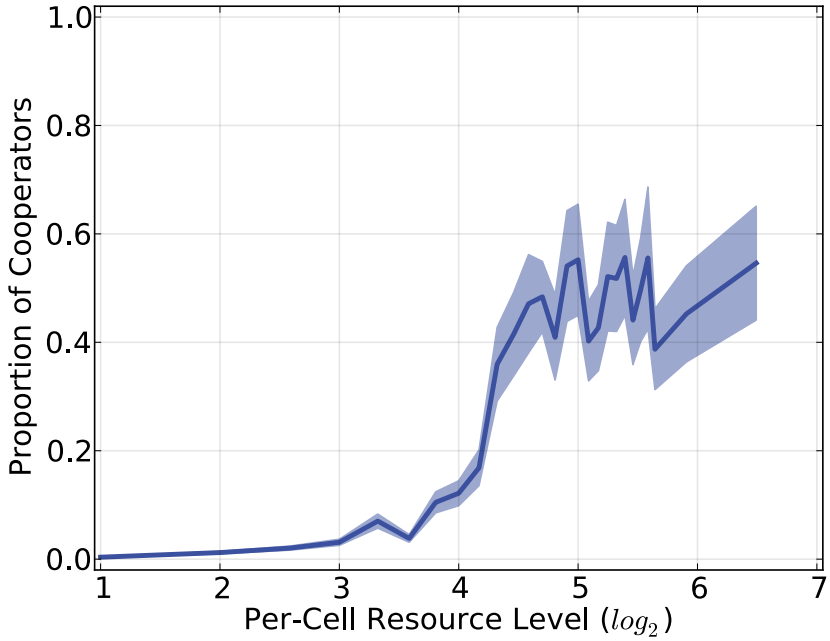



Figure 4.1: Proportion of cooperators in virtual biofilm populations evolved in environments with different resource abundances. Cooperators are defined as those whose phenotypes included the public-good-producing task, and cheaters were those who did not. Shaded region indicates $\pm 1$ standard error.

Hamilton's rule (see Section 2.1), however, such a rapid transition to cooperation makes sense. Hamilton's rule is an inequality that states that a cooperative act can increase in abundance when its associated costs are less than the benefits that it provides multiplied by the relatedness of the individuals involved. Accordingly, it was at that critical point of resource availability in these environments where cooperative benefits began to outweigh the costs, and the inequality was satisfied. After this transition, the proportion of cooperators in populations stabilized at a level common among all resource-rich environments.

These cooperators, which comprised roughly 40–55% of the populations in resource-rich environments, produced enough public good to allow between 70–80% of the cells targeted to stave off the kill event, as shown in Figure 4.2. This success at surviving the disturbance presented by the kill event was achieved by the production of public good, which remained above the kill threshold in resource-rich environments, as shown in Figure 4.3. Although the average level of public good in these environments was similar across populations, its variation among cells was

quite large. Figure 4.4 shows one example of this, where some cells contained up to 30 times more public good than the threshold of the kill event. The inclusion of these cells greatly helps raise the average, and thus can enable even patches of cheaters to survive disturbance.

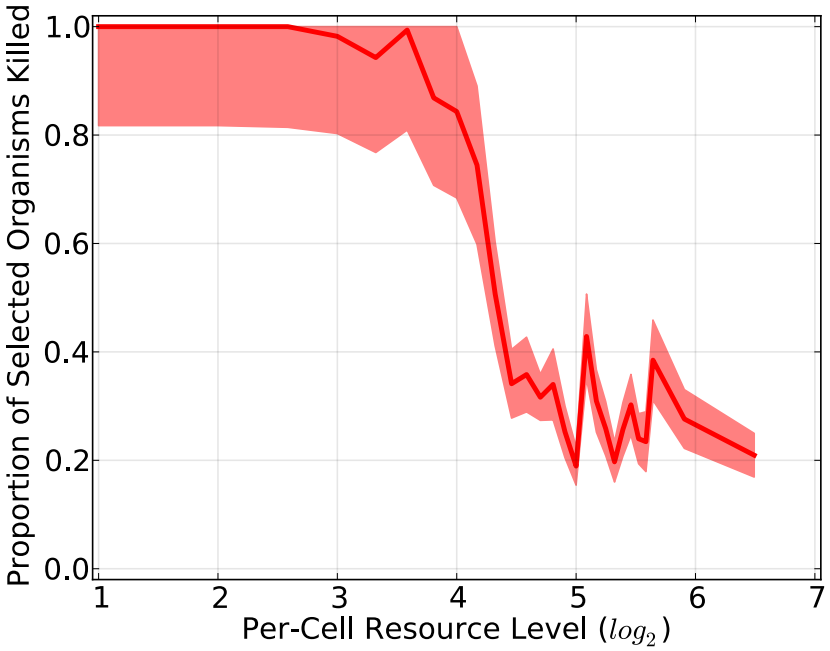



Figure 4.2: Proportion of examined cells killed by the kill event after evolution for 100,000 updates in different resource environments. Shaded region indicates $\pm 1$ standard error.

This variation in public good can be attributed to the spatial structuring of populations. Figure 4.5 shows one representative population in a resource-rich environment. Within these populations, clusters of cooperators formed under the protection from disturbance provided by the production of public good. These clusters of cooperators, however, were continually invaded by neighboring cheaters, which were also protected by the public good, but grew faster. These groups of cheaters would increase in size until the decay of public good made them susceptible to the kill event. Once cheaters in that region had been killed through this disturbance, cooperators could re-invade because of the reduced competition from cheaters. This cycle is shown in Figure 4.6 for a region of cells in one representative population.

Not only did this surpluses of public good allow some cheaters to avoid being killed, it also enabled populations to reach greater sizes, as seen in Figure 4.7. This figure also shows the results of control experiments, where the cooperative task produced no public good, the presence of

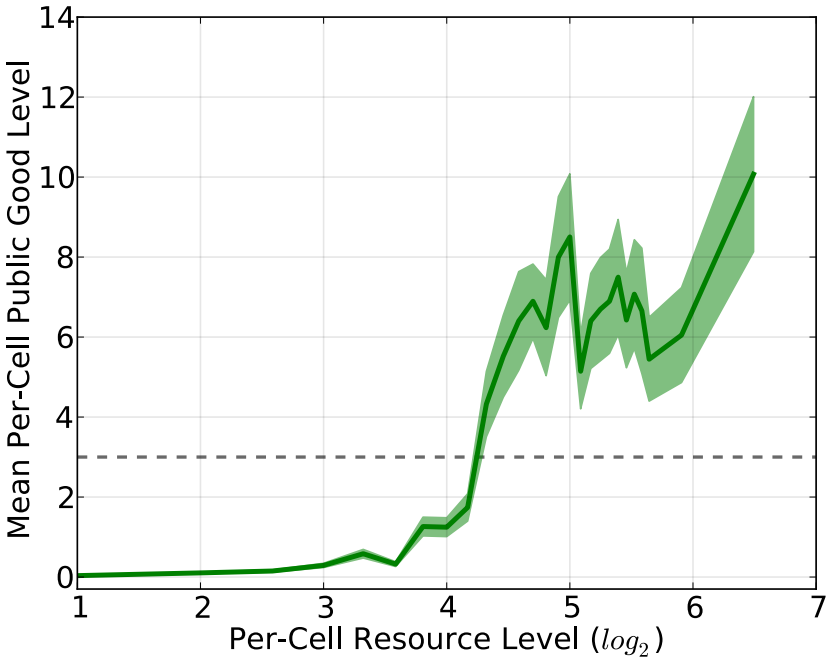



Figure 4.3: Per-cell public good levels after evolution for 100,000 updates in different resource environments. Shaded region indicates $\pm 1$ standard error.

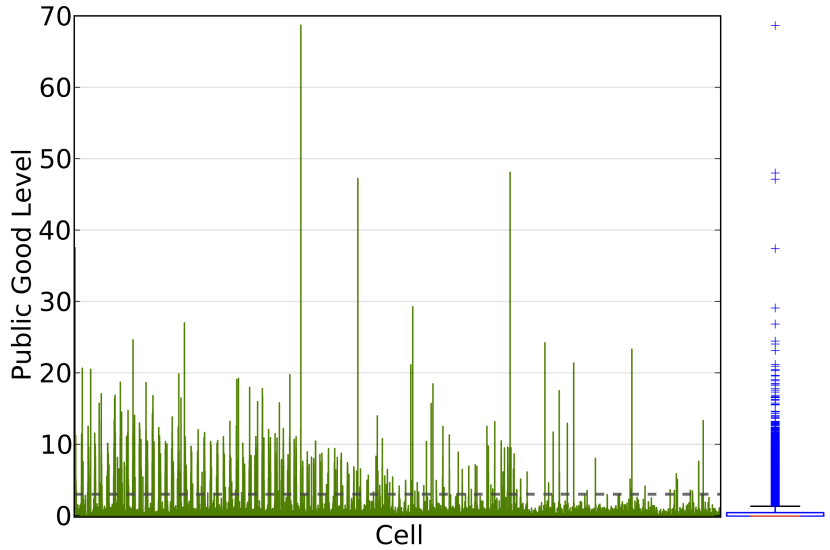



Figure 4.4: Distribution of public good among cells in a typical high-resource environment. A small number of cells have public resource levels much higher than others. The inclusion of such cells in any target area is likely to prevent the kill event from succeeding.

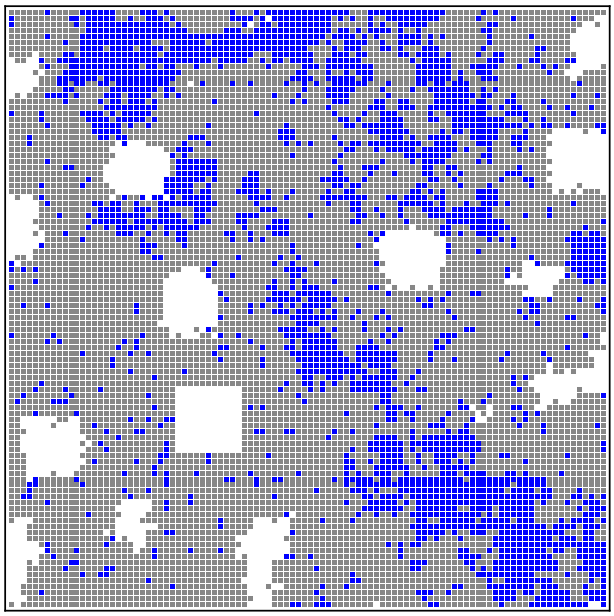


Figure 4.5: Snapshot of a population in the virtual biofilm environment. Here, cheaters (grey) invaded patches of cooperators (blue) until the level of public good in that region had diminished, leaving those cheaters susceptible to the kill event. Empty cells (white) were ideal for cooperators to colonize, as they did not face competition from cheaters in these areas.

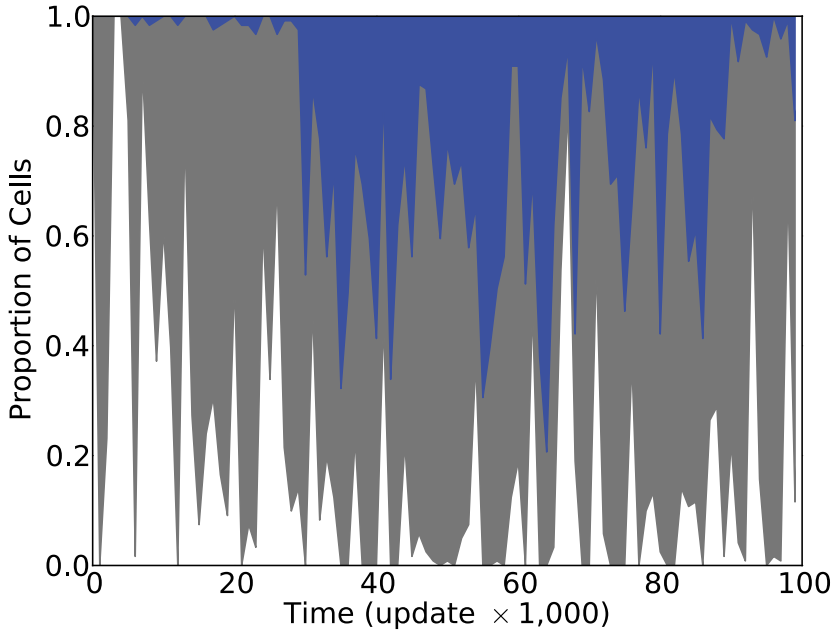



Figure 4.6: Oscillating abundances of cooperators (blue), cheaters (grey), and empty cells (white) present within a selected $5 \times 5$-cell region in one sample population in a high-level resource environment.

cooperators lowered the overall fitness of populations in resource-rich environments, resulting in population sizes that were smaller than in resource-poor environments.

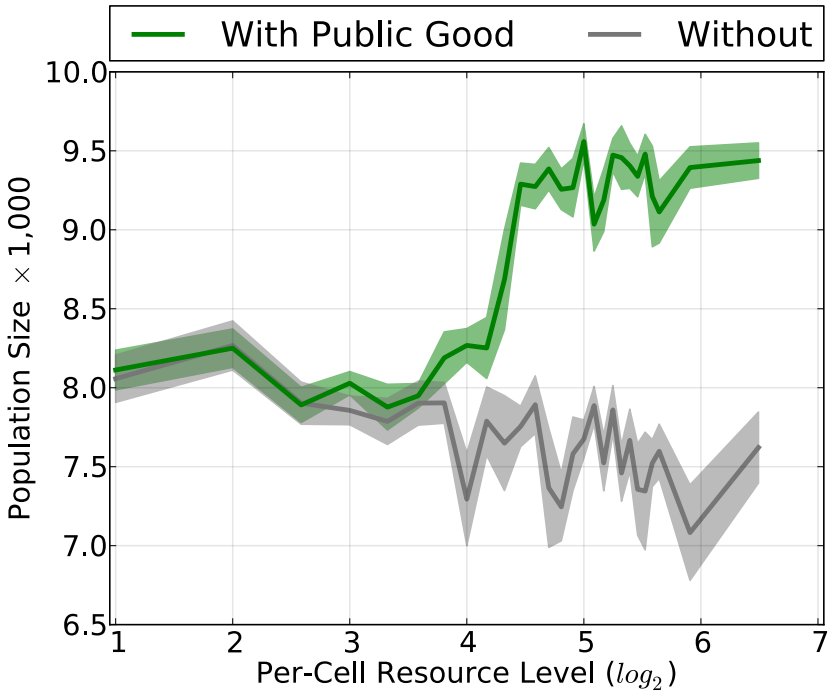



Figure 4.7: Population sizes in the virtual biofilm after 100,000 updates. In resource-rich environments, cooperation enabled populations to maintain significantly larger population sizes. In environments where cooperation did not produce public good, population sizes decreased in these resource-rich environments. Shaded regions indicate $\pm 1$ standard error.

Although cooperators provided these substantial benefits, Figure 4.8 indicates that the cooperative task was only a minor part of the repertoire of behaviors seen in rich environments. As evidenced by the plateau seen in Figure 4.1, cooperation persisted at a minimum level, allowing populations to persist, reach larger densities, and complete greater numbers of growth-oriented tasks.

## 4.2 Biofilm Formation in *Vibrio cholerae*

The previous results offered a new interpretation of into the relationship between resource availability and cooperative behaviors. To further explore these findings and to determine whether the results produced by the virtual biofilm model would accurately capture the dynamics of living

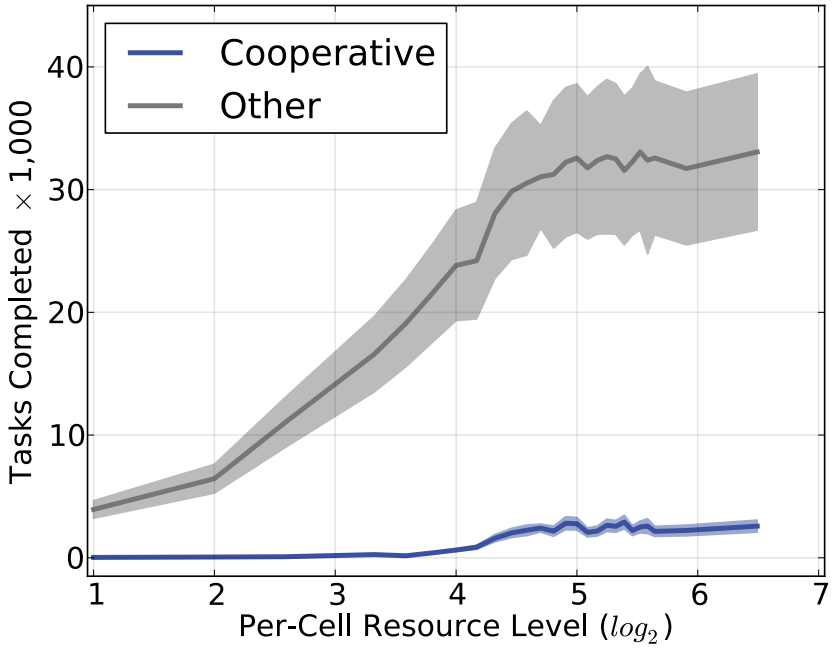



Figure 4.8: Number of logic tasks completed in the last 100 updates by populations evolved for 100,000 updates in different resource environments. Shaded regions indicate $\pm 1$ standard error.

systems, corresponding experiments were performed using populations of biofilm-forming *Vibrio cholerae*.

By placing combinations of these strains in various resource environments, the amount of cooperation in the populations can be inferred from the amount of biofilm produced. In low resource environments, the cheater strain would be expected to outcompete the cooperators, in which case the amount of biofilm produced would be low. However, when resource becomes more abundant, cooperators should be better able to persist, and the amount of biofilm produced should be higher.

### 4.2.1 Materials and Methods

Two strains of *Vibrio cholerae* derived from the wild-type El Tor biotype strain C6706str2 [120, 126] were used to represent cheaters and cooperators. The CW2036 ($\Delta hapR, \Delta vpsL$) strain did not produce biofilm, while the $\Delta hapR$ strain produced biofilm constitutively. Cultures of both strains were started by placing cells from frozen stock in 2 mL $1\times$ LB and incubating overnight with vigorous (220 rpm) shaking at 35 °C

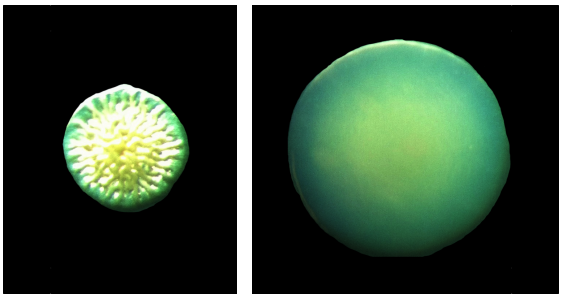
**Growth Media**—Cultures were grown in Lysogeny Broth (LB) (Acumedia Product #7279) [86, 98]. To provide different resource environments, LB concentrations of 0.25×, 0.5×, 0.75×, 1×, 1.25×, 1.5×, 1.75×, and 2× were created by serially diluting a 2× LB stock with a 2× NaCl solution (20 g $L^{-1}$), thereby ensuring an equivalent salt concentration at each resource level.

**Growth of Biofilms**—Wells on a high-throughput Minimum Biofilm Eradication Concentration (MBEC) Assay$^{TM}$system [13] plate containing 145 µL of media were inoculated with 5 µL of bacterial cells comprised of 5:0, 4:1, 1:1, 1:4, and 0:5 ratios of cells from the cheater and cooperator strains. Growth occurred at 37 °C on a Fisher Scientific fixed-speed nutating mixer (22-363-152), which provided gentle rocking. Populations were grown for 6 hours.

**Measuring Biofilm Growth**—After growth, plates were washed with 150 µL phosphate buffered saline (PBS) (pH 7.4), fixed in 150 µL 95% EtOH, and stained with 150 µL of 0.41% crystal violet solution (w/v in 12% ethanol). Following 3 more washes with 150 µL PBS, the crystal violet was diluted in 300 µL 95% EtOH. The absorbance of the diluted crystal violet was then measured at 595 nm using a Molecular Devices SpectraMax M5 plate reader.

**Population Compositions**—In *Vibrio cholera*, biofilm formers display drastically different colony morphologies than do non-formers. Depicted in Figure 4.9, *rugose* colonies (a), which are smaller in size and wrinkly in appearance, are biofilm formers, and *smooth* colonies (b), which are larger and appear more translucent, are non-formers. These morphotypes are determined by a complex hierarchy of regulators [138].

To measure the proportions of cooperators and cheaters present in biofilms, therefore, the relative abundance of these two colony morphotypes was counted. These biofilms were first removed from the plates by sonication for 45 minutes using a Branson Bransonic©2510 at 40 kHz and placed into a new plate containing 150 µL of 1× LB. Afterwards, these solutions were serially diluted in 1× LB to a concentration of $10^{-6}$, plated on LB agar 15 g $L^{-1}$, and grown at 37 °C. The number of rugose (biofilm formers) and smooth (weak biofilm formers) colony forming units were counted after 24 hours. These observed proportions indicated the relative fitness between the rugose cooperative strain and smooth cheaters in the environments in which they were grown.



(a) (b)

Figure 4.9: *Vibrio cholerae* colony morphotypes: (a) The *rugose* colony morphology is associated with biofilm formers, and was seen in increasing abundances in the evolved strains. Rugose colonies are considerably smaller than smooth ones (images are to scale), are more opaque, and appear "wrinkly." (b) The *smooth* colony morphology is associated with biofilm non-formers and was the sole morphotype displayed by the wild type strain. Smooth colonies are relatively large, smooth, and translucent.

As described in Section 3.5.1, increases in the proportion of cooperators indicate that cooperative biofilm formation is favored in that environment, while decreases in this proportion indicate that cooperation was overly costly.

### 4.2.2 Results

Figure 4.10 shows the resulting biofilm production of the mixed populations, while Figure 4.11 highlights biofilm growth in a wider variety of resource environments when populations were begun using a 1:1 mix of cooperators and cheaters. These results show a trend similar to that observed in the digital populations: a rapid transition occurs between low biofilm production in resource-poor environemtns and high biofilm production in resource-rich environments. This transition occurred at resource concentrations between 1.5$\times$ and 1.75$\times$ LB.

Figure 4.12 shows the relative proportions of biofilm formers and non-formers within these biofilms as determined by visually counting the number of colony-forming units of the cheater- and cooperator strains. Although a clear relationship between the number of cheaters present in a biofilm and resource abundance cannot be drawn from these data, it can still be observed that cheaters often comprise a significant portion of the biofilm. These "free riders" have previously

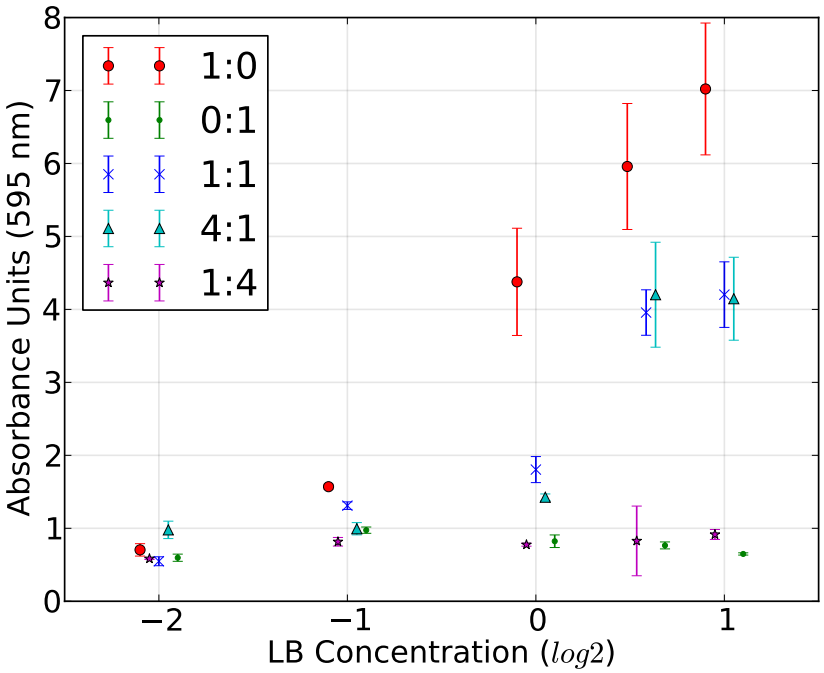



Figure 4.10: Biofilm growth from different starting population compositions (cooperators:cheaters) in different resource environments. When cooperators composed at least 50% of the starting population, biofilm formation increased significantly in resource-rich environments. Error bars indicate $\pm 1$ standard error.

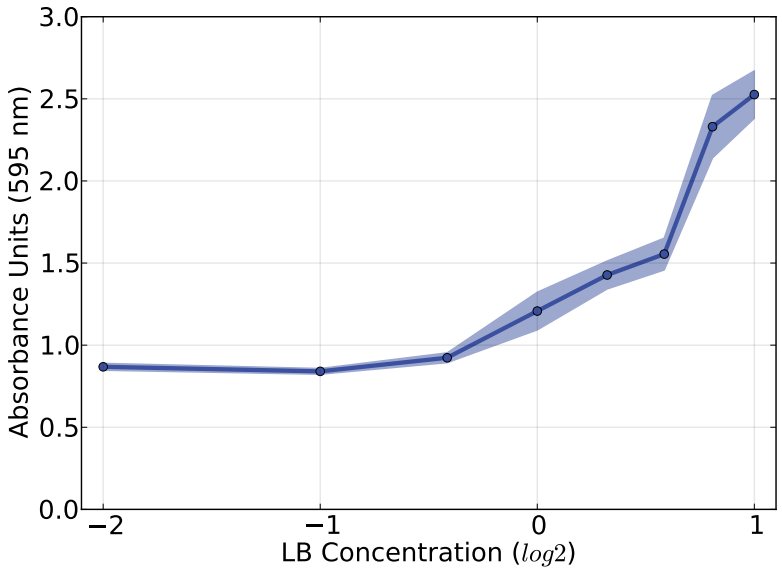



Figure 4.11: Optical density of biofilms grown in different resource environments starting from a 1:1 combination of strains. Shaded region indicates $\pm 1$ standard error.

been shown to weaken biofilms over time as they exploit the EPS produced by others [110].

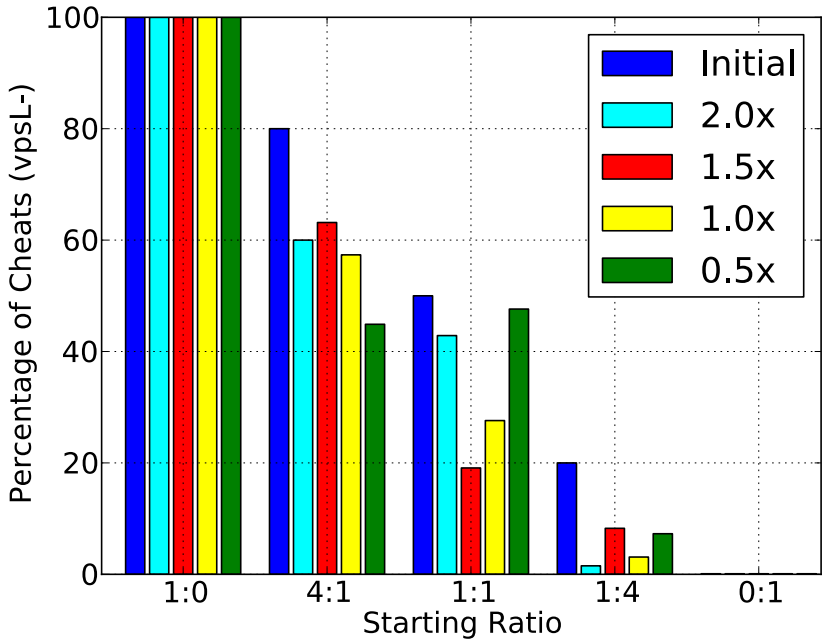



Figure 4.12: Proportion of cheaters present in biofilms after growth in different resource concentrations. Ratios indicate the initial proportions of cheaters:cooperators in the populations.

## 4.3 Experimental Evolution of Biofilm Formation in *Vibrio cholerae*

The experiments described in Section 4.2 showed that cooperation did indeed increase along with increases in resource. However, whether or not a critical transition point at which populations began to favor cooperation similar to that seen in the virtual biofilm environment was not clear from the resulting data.

One explanation for this lack of a clearly-defined transition is that populations simply had not had sufficient time to adapt to this experimental environment. Indeed, the 6 hours, or approximately 2–3 generations, did not offer populations sufficient time for beneficial mutations to arise and have substantial impact on the overall behavior of the population. Although increases in biofilm were seen in all LB concentrations, these increases may not have been stable in lower- and intermediate

concentrations, which would cause each population would adjust over time to better match the environment.

The experimental evolution study presented in this section was conducted to address these potential shortcomings. Populations of *Vibrio cholerae* were grown in similar environments for more than 45 generations in order to more closely approach the levels of cooperative biofilm production appropriate for environments with different abundances of resource.

### 4.3.1 Materials and Methods

Populations of *Vibrio cholerae* strain that could not engage in quorum sensing were grown and evolved in different concentrations of LB, as described in Section 4.2.1. Because this strain cannot engage in quorum sensing, which mediates biofilm formation in *V. cholerae* [57, 126], it forms weak biofilms.

**Experimental Evolution of Biofilm Formation**—Cultures were started by placing cells from frozen stock in 2 mL $1\times$ LB and incubating overnight with vigorous (220 rpm) shaking at 35 °C. For each LB concentration, five replicate populations were created by vortexing a 1:100 dilution of these cultures in the appropriate media. 150 µL of these dilutions were used to inoculate wells in an MBEC$^{\text{TM}}$assay plate. These populations were grown for 24 hours at 35 °C on a Fisher Scientific fixed-speed nutating mixer (22-363-152), which provided gentle rocking. After 24 hours, cells in the biofilm were removed from the plates by sonication for 45 minutes using a Branson Bransonic©2510 at 40 kHz and placed into a new plate containing 150 µL of $1\times$ LB. This product was then diluted using the appropriate media to a concentration of $10^{-2}$, vortexed, and 150 µL was used to inoculate a new MBEC$^{\text{TM}}$plate. The remainder was frozen at $-80$ °C. This process was repeated for 7 days, or approximately 45.5 generations.

### 4.3.2 Results

As Figure 4.13 shows, these populations evolved substantial increases in biofilm formation when evolved in resource-rich environments. However, this increase in cooperation did not occur gradu-

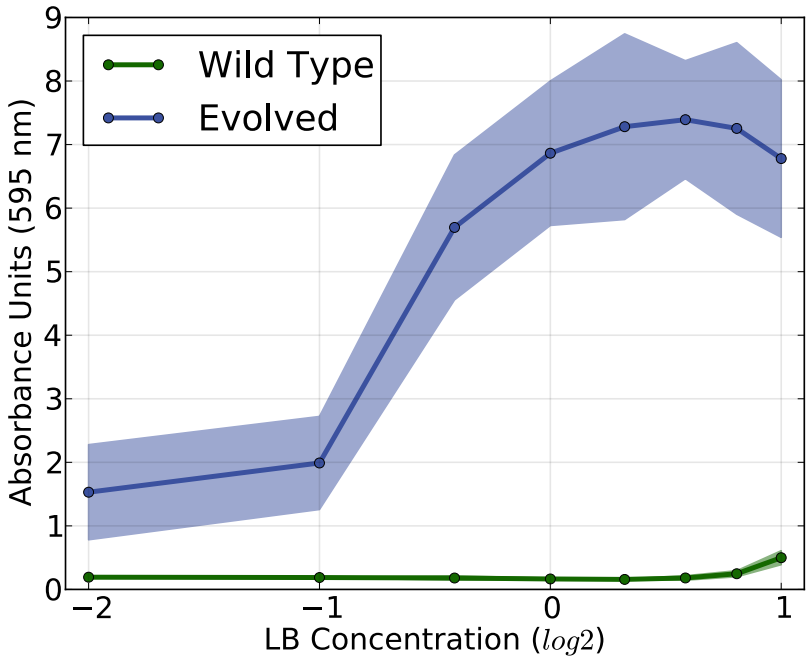



Figure 4.13: Biofilm formation of wild type ancestor strain and 7-day evolved strains. Due to selection for biofilm during passaging, evolved populations in all environments produced more biofilm than the wild type. However, populations evolved in resource-rich environments produced significantly more biofilm than those evolved in resource-poor environments. Shaded regions indicate $\pm 1$ standard error.

ally, as has previously been observed [8], but rather as a rapid transition about some critical level of resource, as shown in Figure 4.13. This critical level occurred between 0.5× and 0.75× LB. Below this level, populations produced uniformly weak biofilms, while above this threshold, uniformly robust biofilms were produced. Although daily passaging selected for biofilm production in all resource environments, the increase seen in resource-rich environments was significantly greater than in resource-poor environments.

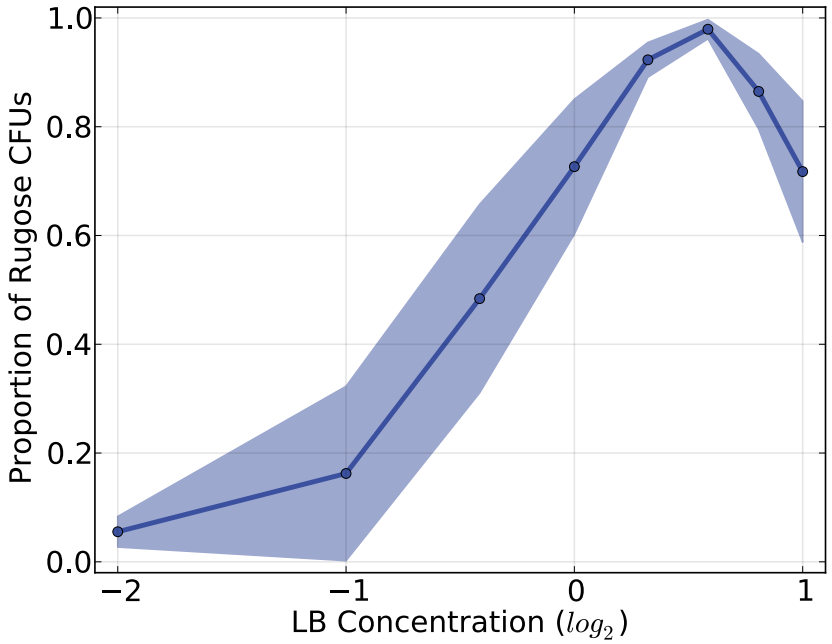



Figure 4.14: Fraction of cooperative (rugose) colony forming units among populations evolved in different concentrations of LB. Shaded region indicates ±1 standard error.

As shown in Figure 4.14, this increase in biofilm formation resulted from increases in the abundance of biofilm-forming rugose colonies within these populations. When plated, constituent cells from the biofilms formed rugose colonies more frequently when evolved in richer environments than in poorer environments. This trend occurred for populations evolved in media up to 1.5× concentration. Beyond this, non-forming rugose colonies began to account for increasing proportions of colonies. This is likely due to increases in "free riders" who take advantage of the large amount of biofilm produced in these populations. Despite this increasing presence of non-formers, the amount of biofilm produced by populations in rich environments was not significantly decreased.

## 4.4 Discussion

The results of these experiments demonstrate that resource abundance can have a significant effect on the evolution and maintenance of cooperative behaviors such as biofilm formation. In both microbial- and digital- populations, a uniform level of cooperation was favored above some critical level of resource. Below this level, populations focused solely on individual growth.

From the perspective of inclusive fitness and Hamilton's rule, described in Section 2.1, such a rapid transition makes sense; it was at that critical point of resource availability in our models where the cooperative benefits began to outweigh the costs, and the inequality was satisfied.

With this stabilization in cooperative output, both models offered evidence that the formation of biofilm through public goods cooperation yielded diminishing returns [43]. More specifically, once a biofilm had formed, the production of any additional EPS did not provide any significant increase in benefit, namely the ability for more cells to be passaged in the microbial model or for more cells to avoid being killed in virtual biofilms. At this point, cooperation became costly, and non-formers were favored. This demonstrates how the fitness of a cooperative strategy can vary depending on environmental conditions.

# Chapter 5

# Social Structure

As discussed in Chapter 2, maintaining spatial relationships within populations can significantly affect the evolution of cooperation. With spatial structuring, the benefits of cooperation are more likely to affect kin, which are generally located nearby. However, as the density of a population increases, perhaps due to the production of biofilm, as discussed in Chapters 3 and 4, competition among kin may increase to a point where cooperation is no longer beneficial.

Although these effects of spatial structuring have frequently been demonstrated, few studies have addressed how variations in the number and distribution of social interactions affect evolutionary dynamics.

The work presented in this chapter examines how the social structure, or the number of neighbors with which an organism interacts, affects a population's ability to maintain diversity. A diverse population is one in which multiple strategies can coexist while competing for the same limiting resource [121], such as cooperators and cheaters in the context of this dissertation. Additionally, such populations are more likely to allow for a rare mutation, such as a cooperative act, to persist long enough to reach a significant frequency. It has previously been demonstrated that diverse populations of cooperators are less likely to be exploited by invading cheaters [9].

Bacterial allelopathy is a natural phenomenon that has previously been modeled to study the effects of spatial structure and cooperation [27, 67, 72]. Localized interactions have been shown to

contribute significantly to coexistence of multiple strains in this system. In these systems, bacteria produce *bacteriocins*, which are toxins that cause surrounding cells that do not express resistance to lyse. In the process, the toxin producer is killed. This act, however, makes the newly-freed space and resources available to neighboring cells (ideally, the kin of the producer). Toxin production is genetically-linked to resistance, so *producer* strains are also resistant to the toxin they produce. It is possible, however, to evolve resistance independent of production. Because such *resistant* strains do not pay the cost associated with production, they are able to grow faster than producer strains, while still maintaining their immunity. These strains, however, grow more slowly than a *susceptible* strain that neither produces toxin nor is resistant. Therefore, in the absence of toxin, a resistant strain will be outcompeted by a susceptible strain. This combination of three strategies is considered a *non-transitive* system in which each strain dominates one strain, but is dominated by the third. These dynamics are captured in the classic rock-paper-scissors (RPS) game, where rock crushes scissors, scissors cuts paper, and paper covers rock.

A major contribution of the work presented here is a new model for representing spatial interactions that more closely match those seen in natural populations. Traditionally, spatial models of this system have used cellular automata on lattices containing a fixed number of vertices, or *cells*, distributed uniformly in space [40, 71]. Within these lattices, each cell is connected to its eight nearest neighbors. To prevent boundary effects, periodic boundaries are commonly used, which form a toroidal grid by creating edges between cells on the periphery of the graph. This results in regular graphs in which each cell has the same number of neighbors, and the distance between any cell and its farthest neighbor is the same for all cells. This regularity indicates that any cell in the grid interacts with exactly as many other cells as any other cell. Further, this distance property indicates that no matter where a dominant strategy begins, it must interact with the same minimum number of cells in order to spread throughout the population.

In this chapter, the role of social structure in the maintenance of diversity is examined by studying a model of this non-transitive system on graphs with differing vertex degrees, and hence different patterns of social interactions. The terms spatial- and social structure are used inter-

changeably, as an organism's potential social interactions are limited to its neighbors in space. The objective is to observe the dynamics of populations on the continuum of connectivity between the regular graphs used in lattice models and well-mixed populations in order to determine at what point diversity is not maintained.

These observations are made using three models. First, lattices are used in Section 5.3, and the number of interactions is increased by expanding the radius of interactions surrounding each cell. This model yields high-level insights into the social structures in which diversity can be maintained. To achieve a more fine-grained control over a cell's interactions, a method for creating graphs of varying expected vertex degrees from a set of points in Cartesian space is utilized in Section 5.4. Diversity is also examined on small world graphs in Section 5.5, where interactions are primarily localized, with the exception of a small number of long-range interactions. Additionally, the importance of population size and the resulting graph connectivity are shown in Section 5.6 to play a critical role in determining whether or not diversity can be maintained. Finally, Section 5.7 demonstrates the stabilizing effect that novelty can have in populations where spatial and temporal variations in fitness exist, such as those targeted in this dissertation.

The spread of a two-strategy system on graphs with differing properties was previously studied by Ohtsuki et al. [96], who formulated a simple rule for the maintenance of diversity. This rule, shown in Equation 5.1, was described by the authors to be equivalent to Hamilton's rule, as connectivity ($k$) can be seen as an approximation for the inverse of relatedness.

$$b/c > k \tag{5.1}$$

The work presented in this chapter differs in that a three-strategy, non-transitive system is used, and the benefits of a particular strategy are not fixed, but rather depend on the composition of each cell's neighborhood. More similar to this work, Károlyi et al. [69] studied increases in social interactions through imperfect mixing of the spatial structure on a lattice. The primary difference is that their work used some measure of mixing, while the work presented here maintains persistent neighborhoods while varying the number of potential interactions. Finally, Buckley

and Bullock [11] used an information theoretic approach to investigate how space contributes to the complexity of a system. Although the focus of their work was different, they showed that complexity can play a large role in a population's ability to maintain diversity.

## 5.1 Cellular Automaton Model

A graph-based cellular automaton model was developed using the SEEDS platform in order to study the effects of social structure on diversity [23]. This model consisted of *cells* that were connected to each other by undirected edges, making both cells neighbors of each other. Interactions in this system were limited to a cell and each of its neighbors. In all experiments described, populations consisted of 90,000 cells. Each cell was either empty or exhibited one of three possible strategies:

1. *Susceptible* cells produced no toxin, nor were they resistant to toxin production by neighboring cells. Because susceptible cells did not pay a cost to maintain either of behaviors, their growth was faster than other strategies.

2. *Producer* cells produced a toxin that could kill neighboring susceptible cells. Additionally, since resistance is a trait that is genetically linked with production in natural bacteriocin systems, producer cells were also resistant to toxin produced by other producer cells.

3. *Resistant* cells can be viewed as producers that cheat. They reaped the benefits provided by adjacent producer cells without themselves paying the costs of toxin production. As such, they exhibited faster growth than producer cells, but slower growth than susceptible cells, due to the added cost of maintaining resistance.

4. *Empty* cells had no effect on their neighbors. When chosen, an empty cell adopted the strategy of a randomly-selected neighbor.

In this chapter, these cell types are referred to as "strategies"; however, they can also be viewed as species, strains, or subspecies. At the beginning of each experiment, cells were randomly as-

signed to be either empty or to exhibit one of these strategies. In this system, the loss of one strategy will break the non-transitive relationship between the strategies, which quickly leads to the loss of a second strategy. As an example in the rock-paper-scissors game, if no paper remained, rock would quickly outcompete scissors, as rock would no longer face competition. Alternatively, if scissors were lost, paper would dominate rock.

Importantly, the growth of each strain was controlled by its rate of mortality. All strategies shared an intrinsic death rate, and the costs associated with resistance and toxin production manifested themselves as increases in death rate. This means that at any given time, a producer cell was more likely to die than a resistant cell, and a resistant cell was more likely to die than a susceptible cell. When a cell died, it became empty. For a cell to change from one strategy to another, it had to first die, and then later adopt a neighboring strategy.

Populations were run for 10,000 time steps, or *epochs*. At each epoch, 90,000 cells were chosen at random, and their states were updated asynchronously according to the rules described below. Following the configuration adopted by Kerr [71], the probabilities of a resistant or producer cell dying during one of these updates were 0.312 and 0.333, respectively. Because the fate of a susceptible cell was tied to the presence of neighboring producer cells, its chance of death, $\Delta_S$, was modeled according to Equation 5.2, where $\Delta_S^0$ is the intrinsic death rate for susceptible cells (0.250 in this work), $\tau$ is the toxicity of producers (0.65), and $f_p$ is the fraction of producers in the cell's neighborhood. All key parameters for this experiment are listed in Table 5.1.

Table 5.1: Bacteriocin Model Parameters

| Property | Value |
|---|---|
| Population Size | 90,000 |
| Replicate Populations | 20 |
| Epochs | 10,000 |
| Intrinsic Susceptible Mortality ($\Delta_S^0$) | 0.250 |
| Resistant Mortality ($\Delta_R$) | 0.312 |
| Producer Mortality ($\Delta_P$) | 0.333 |
| Producer Toxicity ($\tau$) | 0.65 |

$$\Delta_S = \Delta_S^0 + \tau f p \tag{5.2}$$

Studies examining the maintenance of cooperative behaviors commonly compare the fitness costs of a strategy with the benefits it provides (e.g., [2,96]). In the model presented here, the costs can be viewed as the increase in mortality seen by resistant and producer cells. In this sense, the cost of each strategy is fixed and continually incurred. However, due to the spatial nature of this and most other biological systems, the benefits depend on the current distribution of strategies in a cell's neighborhood, and will therefore follow a trajectory similar to that shown in Figure 3.15. For example, toxin production may be highly beneficial when surrounded by susceptible cells, but has no benefit when all neighbors are producers. Likewise, resistance is beneficial in the presence of producer cells, but not in the presence of susceptible or resistant cells.

The following sections describe experiments conducted using this model on different types of graphs, focusing on the number of interactions an organism has and its effect on a population's ability to maintain diversity. First, the metrics used to compare these different types of graphs are explained. For each of the treatments described thereafter, 20 replicate populations were studied. Each population started with a different random seed, which led to differences in the overall structure of the graphs used in the Cartesian and small world treatments, the initial distributions of strategies, the stochastic processes of cell death, and the selection of random replacements for empty cells. These differences allowed populations to follow different trajectories.

## 5.2 Graph Metrics

Because connectivity can vary dramatically among graphs, some properties needed to be controlled in order to maintain similarity and allow comparison among the individual experiments. Although many metrics exist that describe the connectivity of a graph, the clustering coefficient and diameter, both of which are described below, were selected to allow for comparison among the graphs used in this work. These metrics were measured using the NetworkX package [51].

The *local clustering coefficient* of a particular cell measures the connectivity of that cell in its particular network [128]. Specifically, the *clustering coefficient* $C_i$ is defined in Equation 5.3, where $i$ is the cell (vertex) in question, $k_i$ is the number of neighbors of $i$, $N_i$ is the set of $i$'s neighbors, and $E$ is the set of edges in the graph.

$$C_i = \frac{2|\{e_{jk}\}|}{k_i(k_i-1)} : v_j, v_k \in N_i, e_{jk} \in E \tag{5.3}$$

A local clustering coefficient of 0 indicates that none of a cell's neighbors are connected to each other, while a clustering coefficient of 1 indicates that all of a neighbor's cells are connected to one another. The clustering coefficient of a graph is defined as the average of the local clustering coefficients of its cells. This property is relevant in this system, as an area with a higher clustering coefficient allows for indirect interactions, such as "the enemy of my enemy is my friend."

Finally, the *diameter* of a graph is defined as the longest shortest path between any two cells. The diameter therefore provides an indication of how long it would take for a dominant strategy to spread to all cells in the graph. As demonstrated in Section 5.6, a graph's diameter can have significant effects on the dynamics of populations.

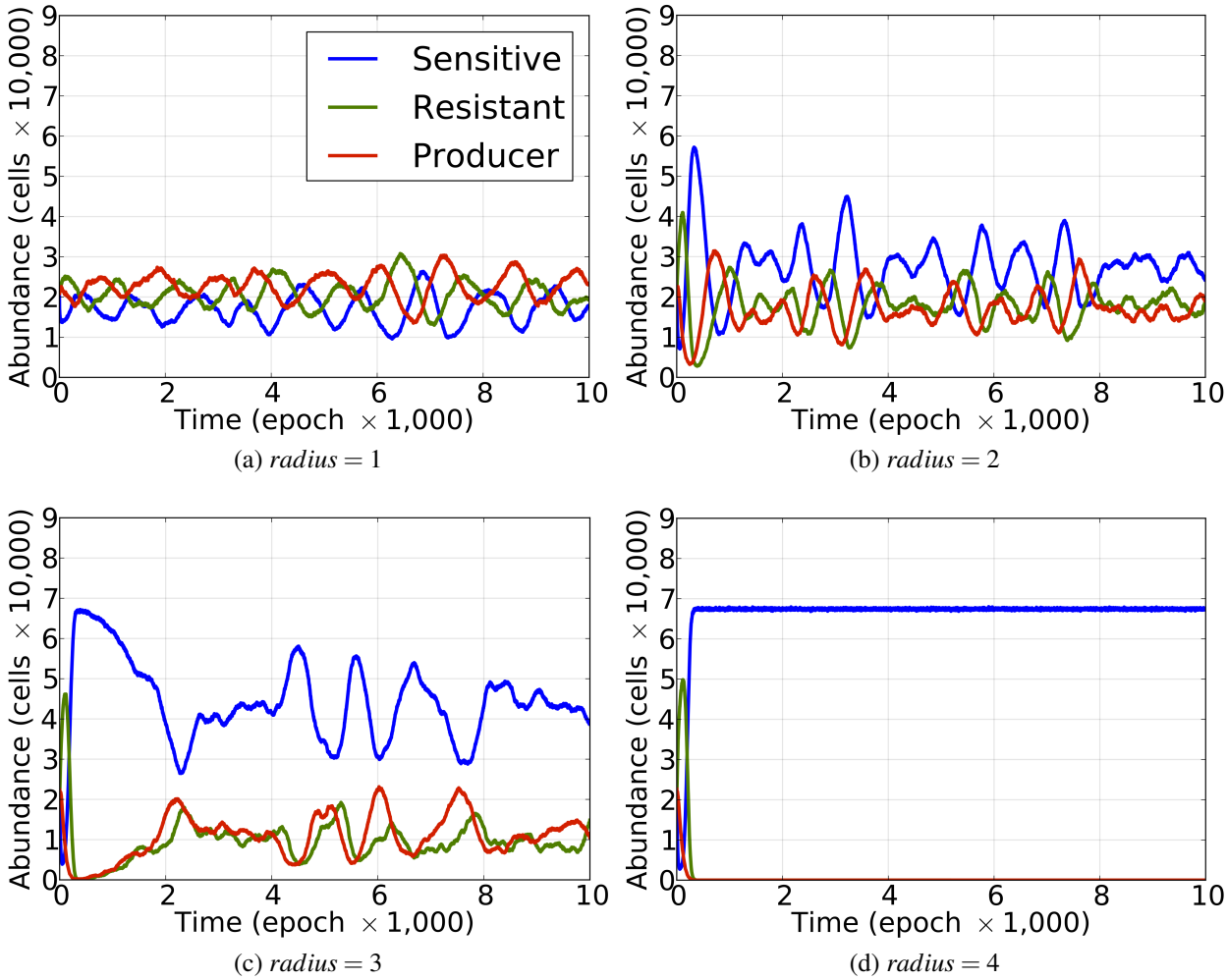
## 5.3 Lattice Model

To examine the effects of increasing social interactions in populations, a lattice model was adopted similar to that used in previous studies (e.g., [27,67,71]). In this model, 90,000 cells were arranged in a $300 \times 300$ grid, with each cell interacting with its 8 surrounding neighbors. Periodic boundary conditions were used, producing 8-regular graphs.

As a simple method for expanding a cell's neighborhood, the radius of interactions was increased. That is, with radius 1, a cell was connected to its 8 surrounding neighbors. With radius 2, a cell's neighbors were the 24 cells within a 2-hop radius. This process continued with increasing radii until diversity was no longer maintained in the populations.

As the radius of interaction was increased, diversity quickly diminished. As the sample popula-

tions in Figure 5.1 show, populations were able to maintain all three strategies when the radius was 1 or 2. At radius 3, several populations were unable to maintain all three strategies, while none maintained diversity at radius 4. As is common in this type of system, in cases where all three strategies coexisted, the strategies persisted in patches, as is shown in Figure 5.2, which moved throughout the environment.




(a) *radius* = 1 (b) *radius* = 2

(c) *radius* = 3 (d) *radius* = 4

Figure 5.1: Strategy counts over time for various neighborhood sizes from representative sample populations. All three strategies remain in all populations when neighborhood radius is 1 (a) or 2 (b). At radius 3 (c), diversity was maintained in 13/20 populations, one of which is shown, while diversity did not persist in any populations at radius 4 (d).

Table 5.2 illustrates the effect of increasing the radius of interactions on the structure of the resulting graphs. The sharp decrease in diameter allows a faster-growing strategy to spread more

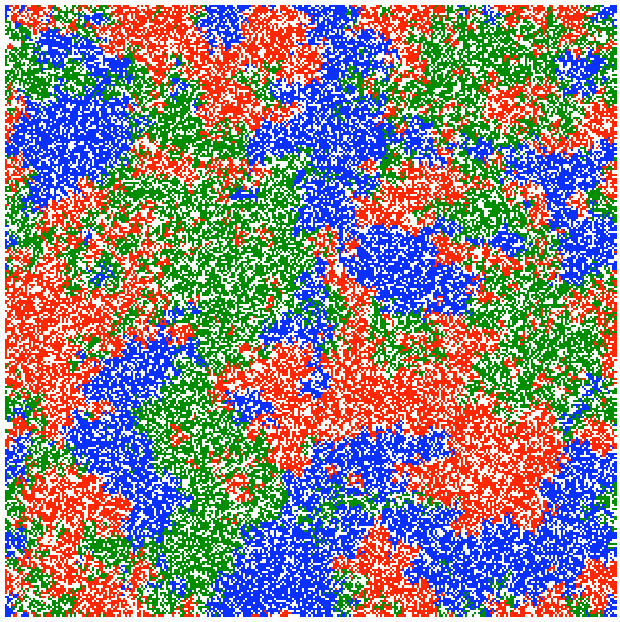


Figure 5.2: Spatial patterns observed in typical populations. When diversity is present, strategies exist in clusters. Sensitive cells are colored blue, resistant are green, producer cells are red, and empty cells are white.

quickly, outcompeting competitors regardless of their capabilities. This effect is exemplified in Figure 5.1d, where the sensitive strategy is seen to quickly eliminate the other strategies.

Table 5.2: Properties of Lattice Graphs Studied

| Neighbors | Diameter | Clustering Coefficient |
|---|---|---|
| 8 (r=1) | 150 | 0.429 |
| 24 (r=2) | 75 | 0.522 |
| 48 (r=3) | 50 | 0.543 |
| 80 (r=4) | 38 | 0.551 |

Although these experiments offered insights into the role that the number of interactions has on diversity, the geometric increases in neighborhood size makes it difficult to observe behaviors at a fine level of detail. Figure 5.3 shows the fraction of populations which collapsed to a single strategy as neighborhood size was increased. Because of this coarse-grained sample, obtaining a significant fit for these data points is difficult using this model.

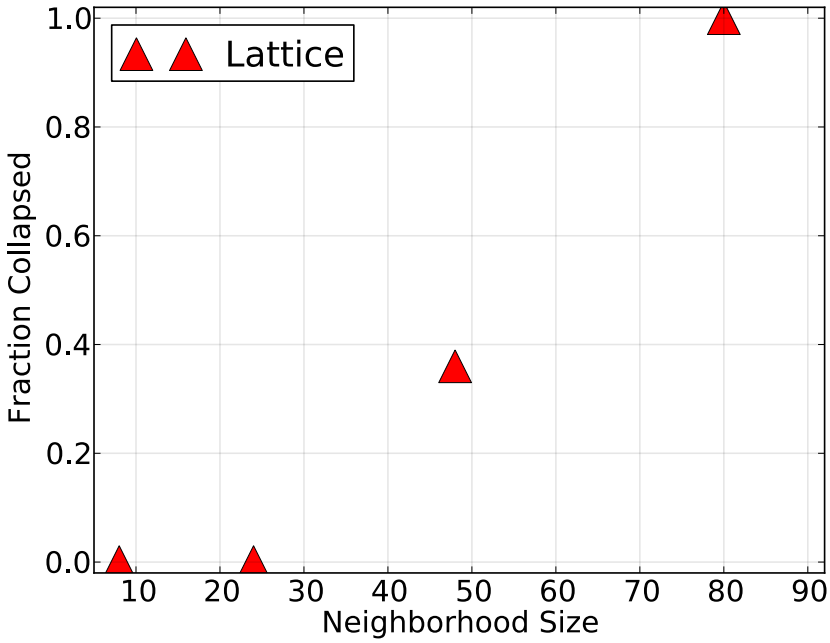



Figure 5.3: Fraction of 20 replicate populations that collapsed to a single strategy for increasing neighborhood sizes.

## 5.4 Cartesian Space Model

Lattice models are well suited for studying spatial effects, but the geometric growth of neighborhood size limits more fine-grained investigation, and is not necessarily representative of natural systems. In order to investigate the effects of more gradual increases in neighborhood size, the next set of experiments moved from lattices to randomly-generated graphs that still accounted for the spatial relationships among cells.

To build these graphs, 90,000 points were randomly placed in a unit Cartesian plane. Each point in this plane represented a cell in the world, and its neighbors consisted of the other points that fell within a circle of specified radius. Since a unit plane was used, the area of the circle was proportional to the expected number of points that it encompassed. That is, the area of a particular circle divided by the area of the plane represented the proportion of points which should, on average, fall within that circle. This construction technique was similar to a previous method that examined how embedding space on random graphs affected various graph properties [3].

The radius that yielded an expected neighborhood size is calculated using Equation 5.4, where

$a$ is the area of a circle, 1 in the left-hand denominator represents the area of a unit plane, $K$ is the expected average number of points within the circle (plus one for the cell the circle is centered on), and $|V|$ is the number of cells in the world, where $|V|-1$ is the number of potential neighbors for a particular cell. Since $a$ is the area of a circle with radius $r$, the radius that will, on average, encompass $K$ cells can be solved according to Equation 5.5.

$$\frac{a}{1} = \frac{K}{|V|-1} \tag{5.4}$$

$$r = \sqrt{\frac{K}{\pi(|V|-1)}} \tag{5.5}$$

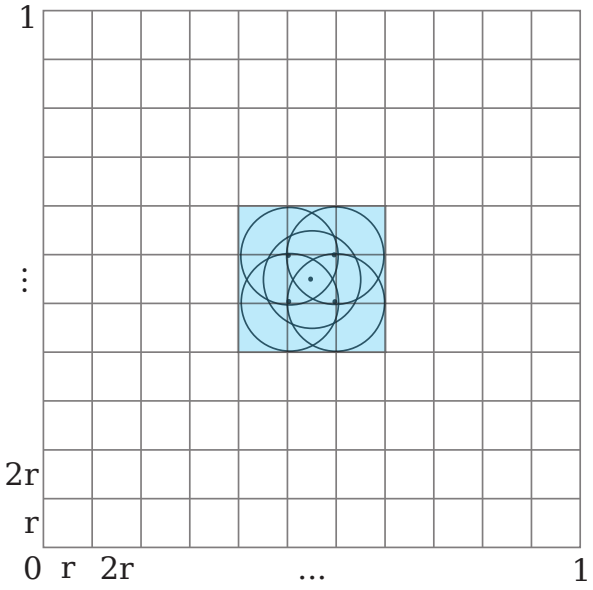



Figure 5.4: Unit Cartesian plane partitioned into bins. Circles show the area where neighbors may fall, and the shaded region is the Moore neighborhood of the central bin.

This treatment also used periodic boundaries, which can be achieved by allowing this circle to wrap around the edges of the plane. To reduce the running time for distance calculations, the plane was partitioned using a grid of two-dimensional *bins*, where each bin contained points that fell within a square area with side length $r$. Since the bins were $r * r$ sized, any point that may have fallen in a circle of radius $r$ around a single point could not be outside of the immediate eight bin

neighbors. Figure 5.4 shows the bin structure overlaying the unit plane and several of the extreme circles with radius $r$, illustrating the fact that all neighboring points must fall within the Moore neighborhood of the bin. This method dramatically reduced the number of points considered as potential neighbors. Additionally, since edges were undirected and the neighbor relation was reciprocal, once the neighbors of a point had been found, that point no longer needed to be considered. This property allowed graph construction to proceed bin by bin, eliminating all points contained within the bin from further consideration after that bin had been examined.

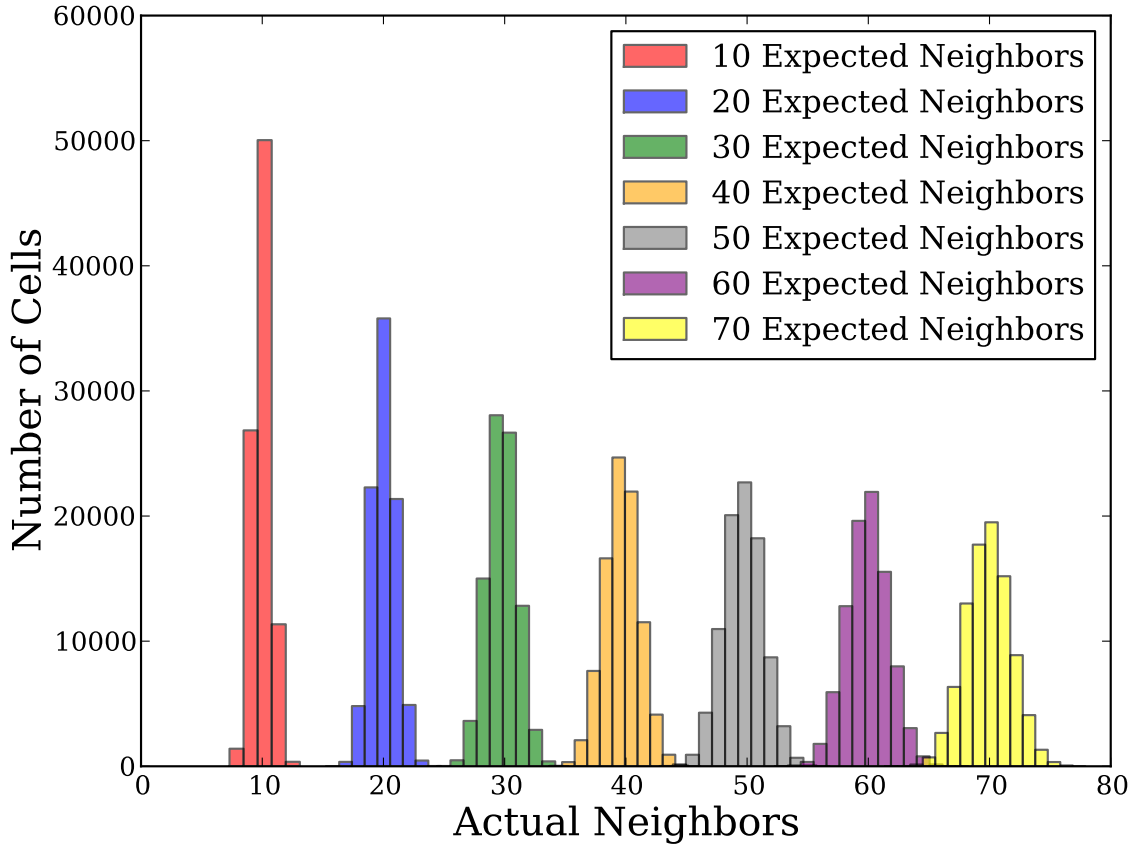



Figure 5.5: Distributions of average neighborhood sizes from 20 replicate graphs with expected neighborhoods from 10 to 70 cells in increments of 10

Figure 5.5 shows the average distribution of neighborhood sizes in sample graphs when varying the expected number of neighbors from 10 to 70. The mean number of neighbors for each treatment was equal to the expected neighborhood size calculated. This method provided fine-grained control over neighborhood size, while maintaining spatial interactions similar to those of lattices.

Random graphs created in this way are arguably more representative of biological systems than lattice models, since some variation is expected in the number of interactions among organisms in the population, even with explicit spatial structuring.

This *Cartesian* method was used to generate random graphs on which to run the cellular automaton model. The properties of the resulting graphs are listed in Table 5.3. It should be noted that several of the graphs generated with expected neighbor size of 10 were disconnected, as one might expect in a natural population with limited interactions.

Table 5.3: Properties of Cartesian Graphs Studied

| Expected Neighbors | Diameter | Clustering Coefficient |
|---|---|---|
| 10* | 45.5 | 0.585 |
| 20 | 83.25 | 0.587 |
| 30 | 57.25 | 0.588 |
| 40 | 51.5 | 0.589 |
| 50 | 59.0 | 0.588 |
| 60 | 53.0 | 0.586 |
| 70 | 49.0 | 0.587 |
| 80 | 45.0 | 0.587 |
| 90 | 38.0 | 0.591 |

The strategy count plots for this topology are similar to those shown in Figure 5.1, and thus are not included. Instead, simplex phase planes were used to examine the dynamics observed in the populations. A simplex phase plane depicts the proportion of strategies that were present in the population at a given time and the trajectory the population took over all. The three corners of the triangle represent the three strategies: producer (P), sensitive (S), resistant (R). The relative distance from each corner depicts the proportion of the population the strategies comprise. Thus, a point in the center of the simplex would have equal frequency of each strategy, and a point at the P corner of the triangle would represent a population completely composed of producers.

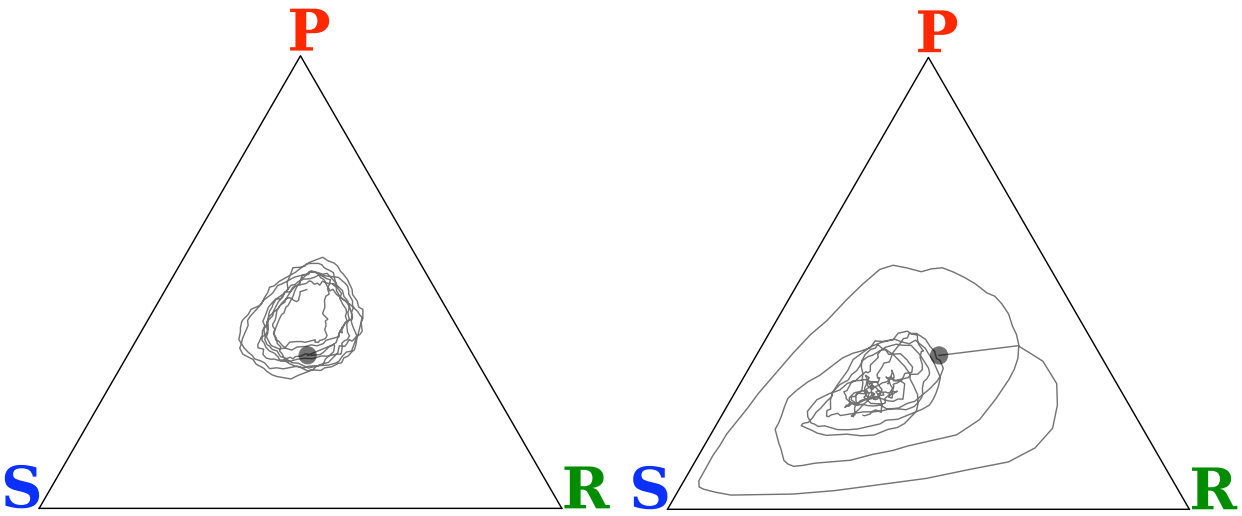
Figure 5.6 depicts four simplex phase planes for different neighborhood sizes roughly corresponding to those examined using lattices. The oscillatory dynamics observed in Figure 5.1 are also present in this topology, and are distinguishable by the circular path within the phase plane in Figure 5.6a. Similarly, the large swings in cell counts with increased neighborhood sizes form the

larger circular paths depicted in Figure 5.6b and 5.6c.

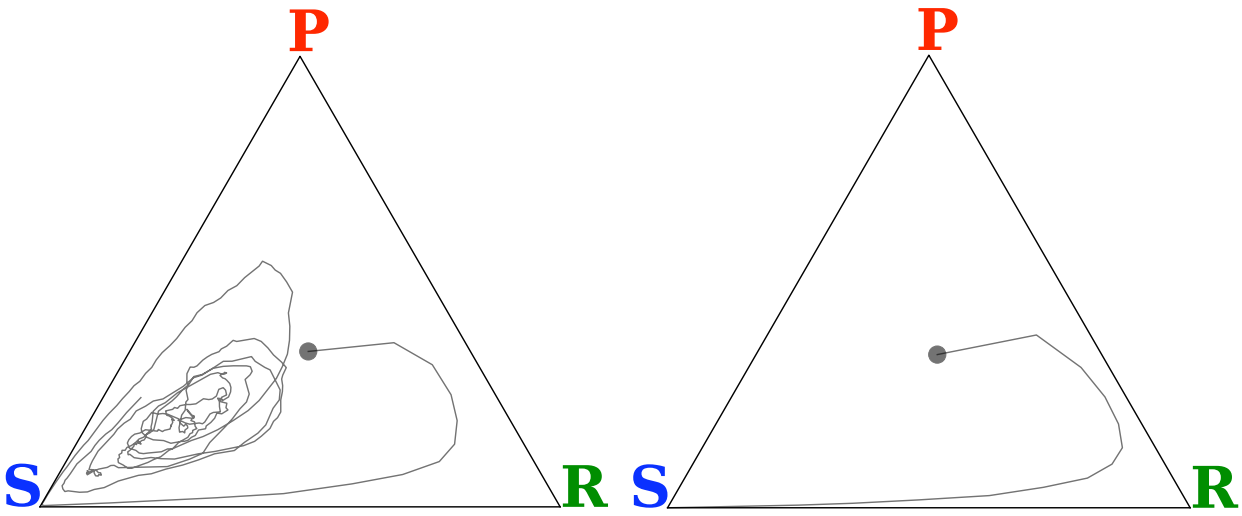
Several populations maintained diversity despite having larger neighborhood sizes (such as that shown in Figure 5.6c) exhibited drastic transient dynamics, where the population of one or more strategies came dangerously close to being eliminated. It is these initial transient dynamics that stochastically led populations to collapse as the mean neighborhood size increases. That is, in those runs that persist through these periods of transient dynamics, the population ends up in a safer region of phase space, one that is less susceptible to stochastic extinction. Of course, as the neighborhood size continues to increase, so does the magnitude of oscillations, and eventually all populations will collapse to a single strategy as the others are driven to extinction, as illustrated in 5.6d.

These transient dynamics were due to initial conditions, in which each cell strategy (including empty cells) was uniformly distributed throughout the world. As depicted in Figure 5.2, clusters of strategies emerged, and it was during the transition between the initial and fully-clustered states that populations often collapsed. Essentially, populations were started in a random state with respect to clusters of strategies. While this approach biases the population towards larger cycles at least initially, it means estimates generated for the collapse of diversity are conservative.

Although the previous figures show the similarities that the dynamics of populations modeled on these graphs had with those on lattices, the Cartesian method allowed this system to be studied in much greater detail. Figure 5.7 plots the proportions of collapsed populations for a range of neighborhood sizes, with the range that produced intermediate loss of diversity serving as a middle point. These data display a rapid transition from populations that can maintain diversity to those that cannot as the number of interactions an organism had was increased. The logistic curve of best fit for these data is highly significant, with an $F$ statistic of 247.62 ($p \ll 0.001$) and an adjusted $R^2$ of 0.985.




(a) 10 neighbors (b) 30 neighbors




(c) 50 neighbors (d) 80 neighbors

Figure 5.6: Simplex phase planes for representative Cartesian topology runs with increasing number of neighbors. The initial distribution of strategies is indicated with a dot. Here, the increases in the magnitude of oscillations are shown as neighborhood size increases. (a)-(b) When neighborhood sizes are small, populations are able to maintain diversity. (c)-(d) However, as connectivity increases, oscillations in strategy abundances can quickly lead to the loss of diversity.

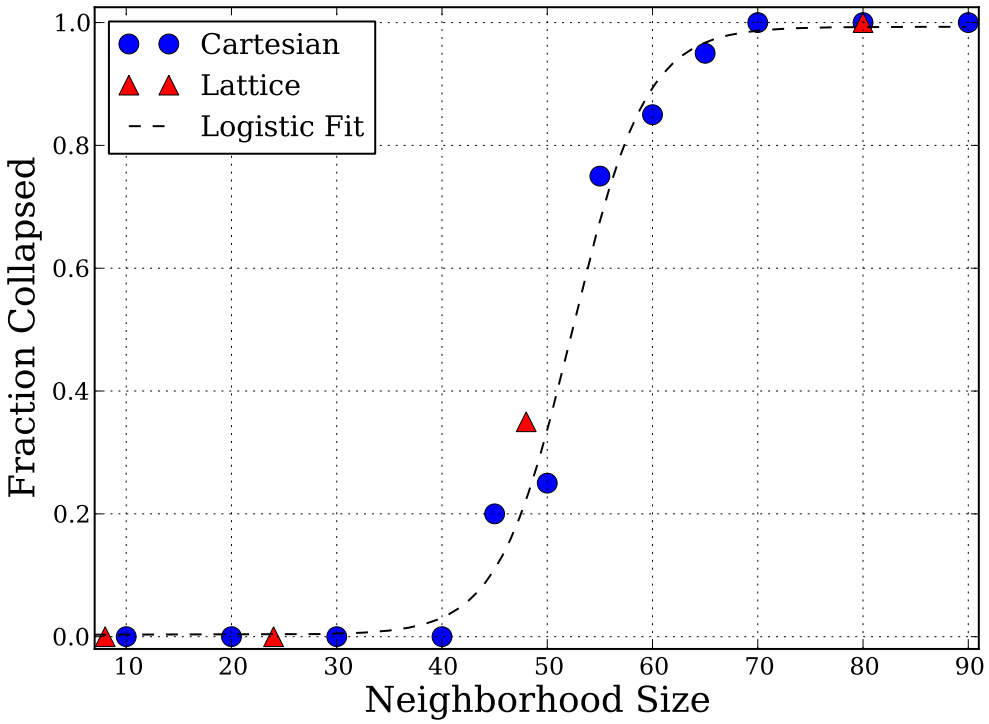



Figure 5.7: Fraction of populations (out of 20) that collapsed to a single strategy across different expected neighborhood sizes - $F$ value 247.62 ($p \ll 0.001$), adjusted $R^2$ 0.985

## 5.5 Small World Networks

Finally, the maintenance of diversity was studied in this system when the interactions among organisms were modeled as *small world* networks, which consist primarily of localized interactions with some long-range interactions [128]. These interactions often result in graphs where the number of interactions separating any two cells is surprisingly small. This property is familiar to those who have played the "Six Degrees of Kevin Bacon" game, where players are able to connect any person to actor Kevin Bacon through at most six social interactions [17]. Although these networks likely do not capture the highly-localized interactions of microbial populations, they have been observed to capture several natural phenomena, such as the gene expression networks [123], and may offer some insight into the maintenance of diversity in the presence of gene flow through these long-range interactions.

To construct these graphs, 90,000 cells were arranged on a ring, and each cell was connected

to its nearest 8 neighbors. For each cell, additional interactions were created by probabilistically adding an edge to a randomly-chosen cell. At probability 0, these graphs were regular and had a diameter equal to the number of cells divided by the neighborhood size. At probability 1, the resulting graphs become random, mimicking interactions in well-mixed populations. For this work, the effect that long-range interactions have on maintaining the diversity of this system are examined.

As shown in Figure 5.8, even a small probability of such interactions had a dramatic effect on the system. Diversity quickly waned when the probability of adding these interactions was between 1% and 2%, which resulted in an additional 900 and 1800 pairs of interactions, respectively, on average. These additional interactions decreased the diameter of the resulting graphs from 11,250 to an average of 54.5 when the probability was 1% and 32.3 when the probability was 2%. The clustering coefficients for these configurations were uniformly 0.631 and 0.620, respectively. The difference in dynamics between systems at 1% and 2% edge creation possibility is shown in Figure 5.9.

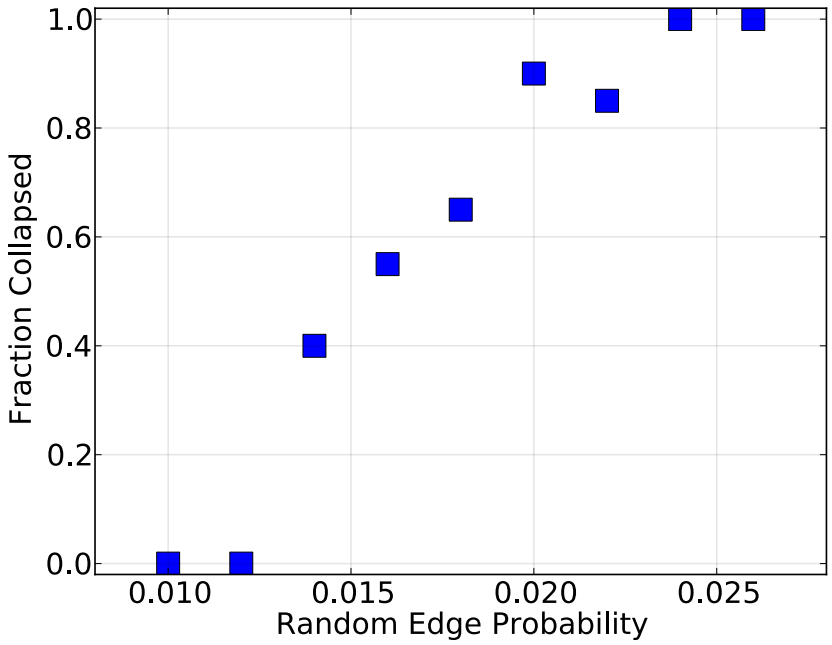
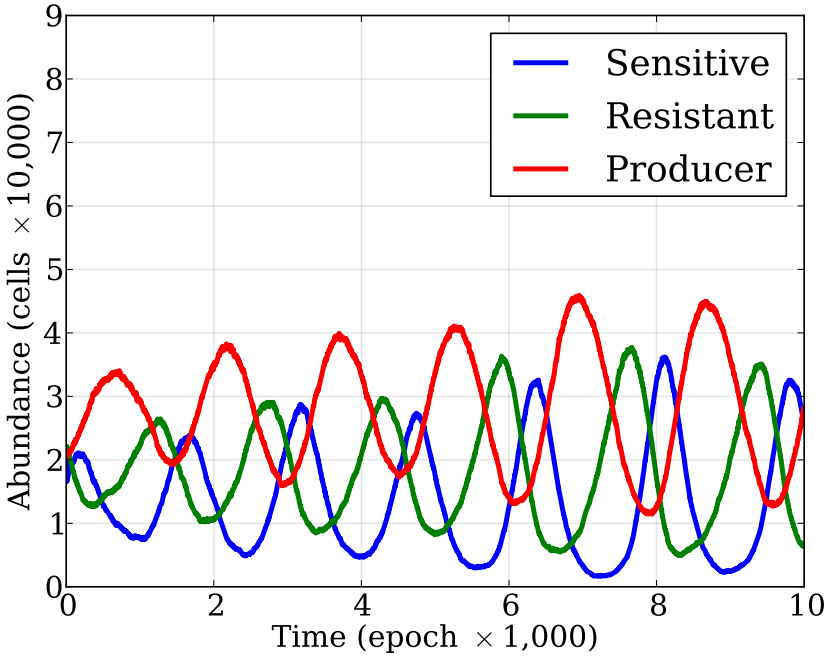
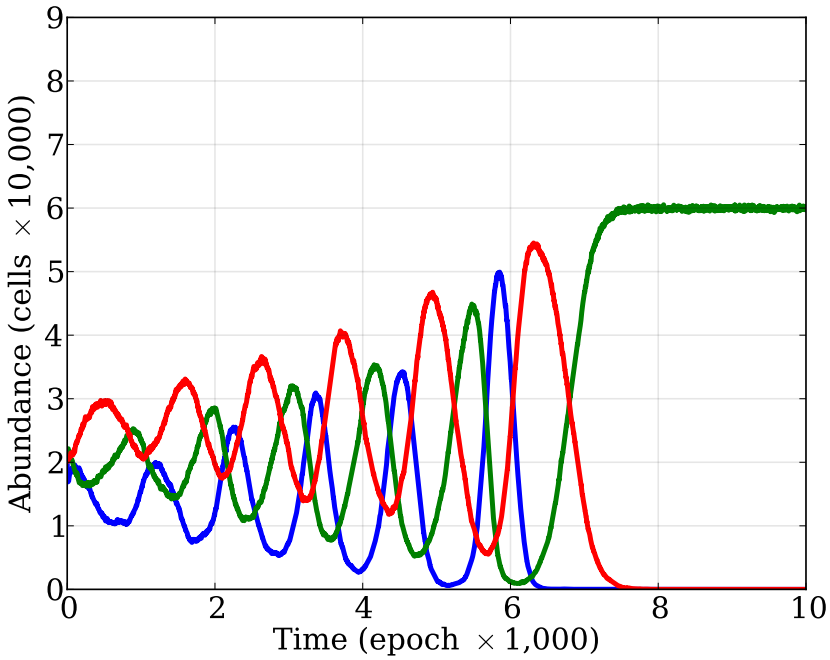



Figure 5.8: Fraction of populations (out of 20) that collapsed to a single strategy in small world networks with increasing probabilities for additional random interactions

Considering the small diameters typical of small world graphs, it is perhaps not surprising that diversity is quickly lost when long-range interactions are added. In the absence of these




(a) $p = 0.01$




(b) $p = 0.02$

Figure 5.9: Strategy counts over time in representative small world networks. (a) At 1% probability of creating a random edge, diversity is maintained. (b) At 2%, diversity is lost.

long-range interactions, the diameter of these graphs was 11,250. Adding additional edges with probabilities between 1% and 2% quickly shrank the diameters in these environments, which made the formation of isolated clusters of strategies difficult. Nonetheless, these experiments provide a dramatic insight into how small increases in interactions can hinder diversity.

## 5.6 Population Size

Each of the experiments described in the previous sections demonstrated that the likelihood that diversity could be maintained decreased along with decreases in the diameter of the network of interactions. In these experiments, decreasing diameters were a result of changing the local neighborhoods of interactions.

To directly address the relationship between diameter and the maintenance of diversity, populations of various sizes were studied using the Cartesian space model discussed in Section 5.4, with an expected neighborhood size of 10 cells. For this experiment, 20 replicate populations with sizes between 1,000 and 20,000 were used. Each topology used for these replicate populations was generated randomly according to the process described in Section 5.4, which was repeated until a connected graph had been created. In populations of 90,000 cells, this topology enabled the maintenance of the three strategies for the duration of the 10,000-epoch experiments in each of 20 replicate populations, as shown in Figure 5.7.

Figure 5.10 shows the proportion of replicate populations that converged to a single strategy for each of the population sizes studied, and Figure 5.11 depicts the corresponding diameters of their interaction networks. These diameters increased linearly with increasing population size, and allowed the majority of populations with at least 8,000 cells to maintain all three strategies. All populations of size 16,000 cells or larger maintained diversity.

These results further demonstrate that the connectedness of populations greatly affects their ability to resist invasion. This occurs when the benefits of any one strategy, cooperative or not, do not allow that strategy to affect all cells in the population before an alternate strategy becomes

favored.

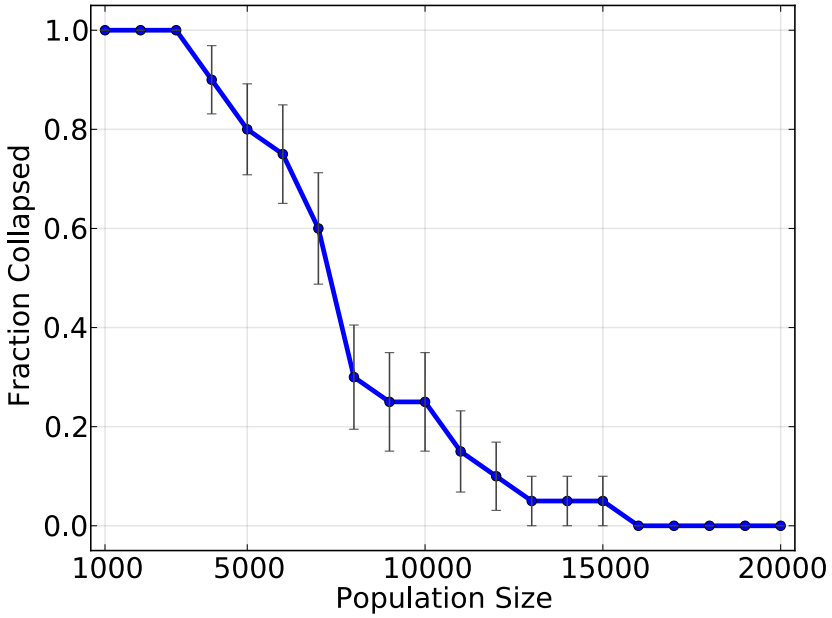



Figure 5.10: Fraction of populations that converged to a single strategy for various population sizes. As populations increased in size, they were more able to maintain all three strategies. Error bars indicate $\pm 1$ standard error.

## 5.7 Novelty and the Maintenance of Diversity

Typically in simple cellular automaton models, such as those used in this chapter, once a strategy becomes extinct, it remains absent from the population. In the case of the non-transitive bacteriocin model used in this chapter, the loss of one strategy results in the loss of a second strategy. As seen throughout this dissertation, however, when the environment changes, either through biotic factors, such as the density of one strategy or niche construction, or through resource abundance or other abiotic factors, selection can change which behaviors are favored.

Therefore, even in populations that have otherwise converged to a single strategy, a source of novelty may allow diversity to be maintained by re-introducing an alternate strategy that would be favored in the new environment. One way in which novelty has been introduced in cellular automaton models is through migration, either at some fixed rate [105] or between connected sub-populations in metapopulation models [90]. Migration, or gene flow, is known to increase diversity

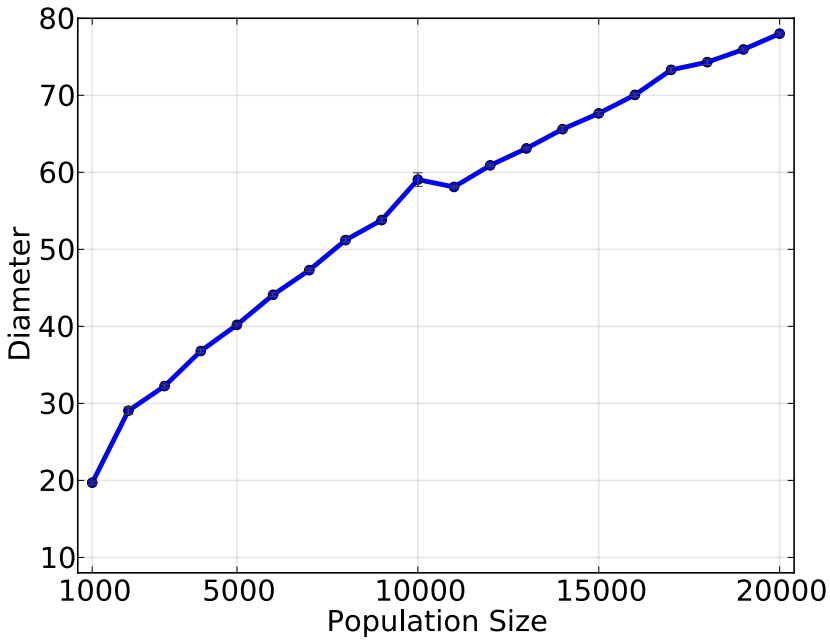



Figure 5.11: Diameters of population topologies for various population sizes. With a constant expected neighborhood size, larger populations have larger diameters, which slows the rate at which a dominant strategy can spread throughout the population. Error bars indicate $\pm 1$ standard error.

within populations and decrease diversity among different connected populations [24, 59], and can lead to significant changes in population dynamics [61]. A second potential source of novelty are mutations, which cause a cell's strategy to differ from that of its parent [88, 105].

To explore the stabilizing effect that sources of novelty can have on diversity, the dynamics of populations using the model of bacteriocigenic bacteria introduced in Section 5.1 were studied in the absence of novelty, with mutations, with migration, and with both sources of novelty. Based on the results presented in Section 5.6, a population size of 5,000 cells was chosen so that populations would be unlikely to maintain diversity in the absence of novelty. For each treatment, described below, 30 replicate populations were used.

Migration events caused a cell to be replaced by a cell of randomly-selected strategy (including empty). For each cell in the population, migration events occurred with probability 0.001 at each epoch. Mutation events occurred when a cell reproduced into a neighboring empty cell. Normally, the offspring inherited the strategy of its parent. When a mutation occurred, however, the offspring's strategy was selected randomly (including empty, which represented a lethal mutation).

These mutation events occurred during replication with probability 0.01.

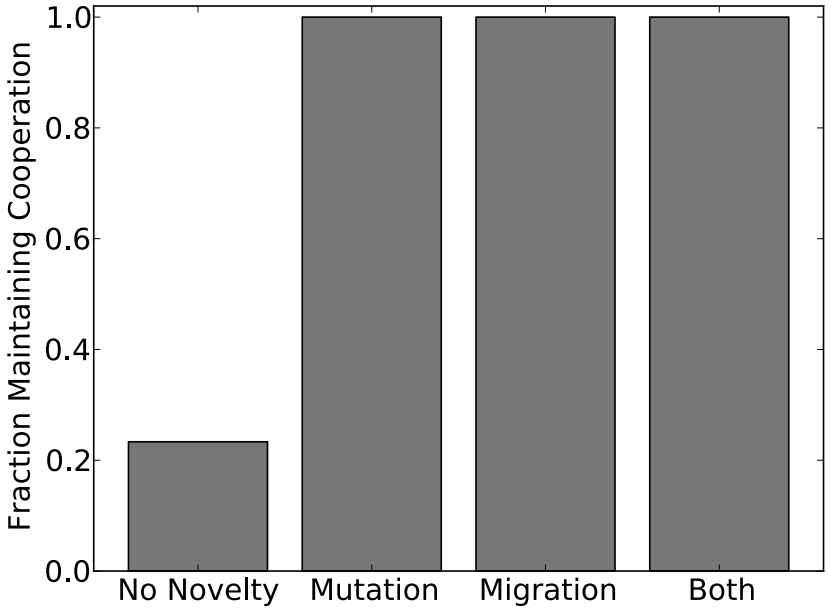
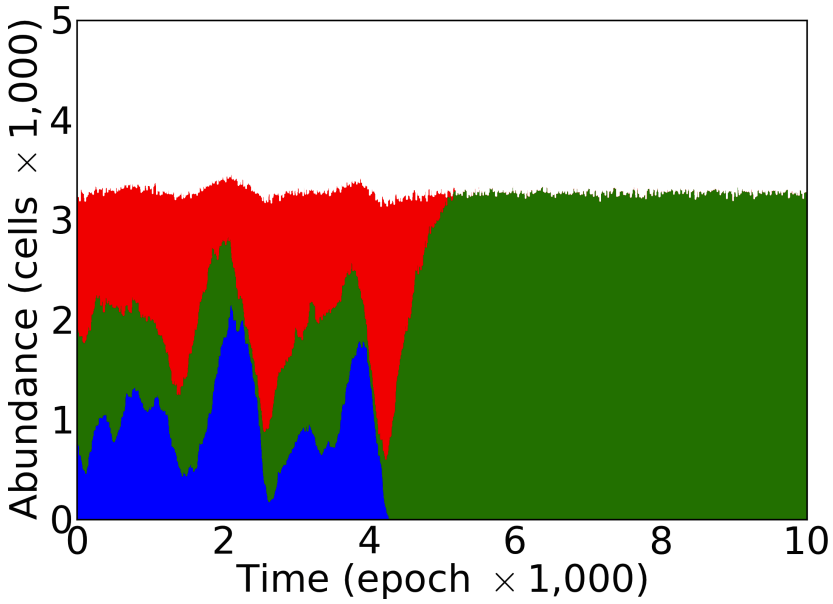


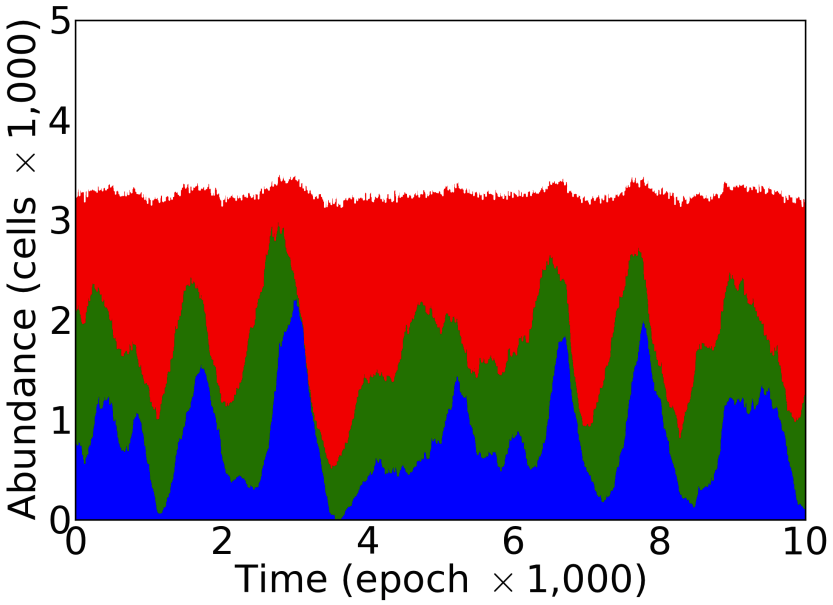

Figure 5.12: Fraction of populations that maintained diversity for various population sizes. Without the ability to re-acquire strategies, only 7 of 30 populations were able to maintain diversity. With sources of novelty, however, all populations were able to maintain all strategies.

As shown in Figure 5.12, when there were no sources of novelty, only 7 of 30 replicate populations were able to maintain diversity. Figure 5.13 shows the distribution of strategies in two sample populations. However, all three strategies were maintained in populations in which mutations and migrations allowed novel strategies to be introduced.

As was the case in previous experiments, the magnitude of oscillations between strategies was often indicative of a population's ability to maintain diversity. Figure 5.14 illustrates these oscillations under the treatments examined. Without novelty, populations showed large oscillations in the abundances of each strategy. In 77% of populations, these oscillations led to the loss of diversity, as shown for one such population in Figure 5.14a. With the introduction of novelty, however, these oscillations were greatly dampened, which allowed all populations to quickly adapt to changing conditions and maintain all three strategies. In almost all cases, the abundance of each strategy was greater than what can be accounted for by mutation or migration alone. This indicates that novelty allowed strategies to be introduced that had a selective advantage. In the environments

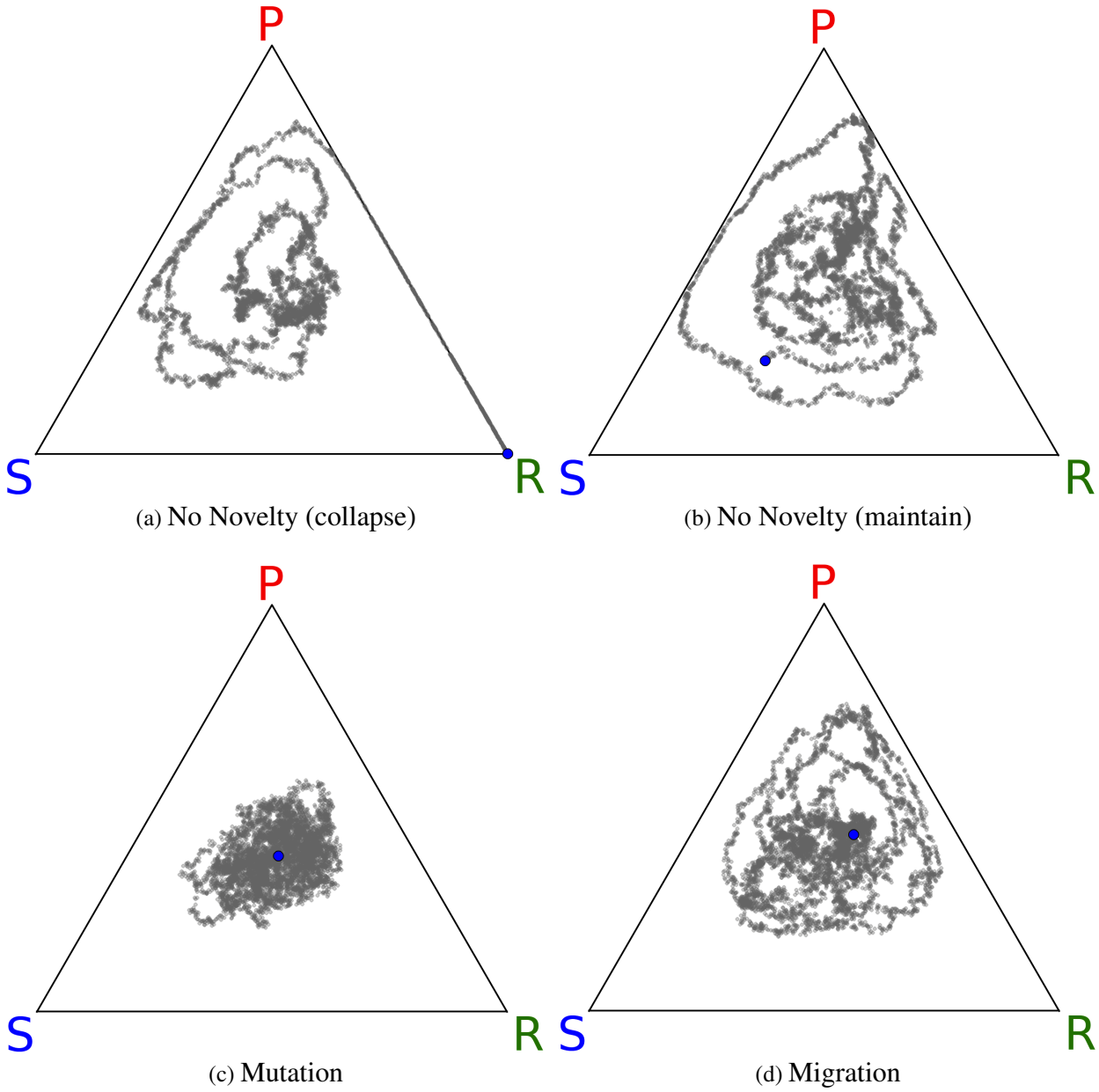



(a)




(b)

Figure 5.13: Strategy abundances in populations with no sources of novelty. Shaded regions represent the relative abundances of sensitive (blue), resistant (green), producer (red), and empty (white) cells. (a) The majority of populations collapsed to one strategy after large oscillations caused one strategy to be eliminated. In this example, producer cells eliminates sensitive cells, but were then outcompeted by resistant cells. (b) Some populations were able to narrowly avoid this fate. In this instance, sensitive cells were nearly excluded near epoch 3,800, yet were able to regain presence.




(a) No Novelty (collapse)

(b) No Novelty (maintain)

(c) Mutation

(d) Migration

Figure 5.14: Oscillations in strategy abundances. (a) In populations with no novelty, large oscillations often caused the loss of diversity. (b) Those that did maintain diversity experienced large oscillations, but narrowly avoided the loss of strategies. (c) With mutations enabled, oscillations were greatly dampened. (d) Immigration of strategies also helped dampen oscillations and enable populations to maintain diversity.

studied, populations in which both sources of novelty were present showed the same dynamics as those that had only mutation or migration. In other environments, however, having both sources of novelty could have a more pronounced effect.

The results presented in this section demonstrate that novelty, introduced either through mutation or migration, can help to stabilize diversity and allow populations to more quickly adapt to changing environments. Because of this key role that novelty plays in evolution, it should be included in all models that aim to accurately capture the evolutionary and ecological dynamics of behaviors within populations.

## 5.8 Discussion

This chapter has demonstrated the effect that social structure, which characterizes the interactions among organisms in a population, has on the maintenance of diversity. This is often important for the evolution of cooperation, as a population that can maintain cooperation is one in which cooperators can resist exclusion by cheaters.

The effect that an individual's neighborhood was examined using a several different graph structures. The results of these experiments showed increases in the number of interactions that each individual in the population has quickly led to the loss of diversity. This loss of diversity was attributed to the diameter of the population's interactions, which represents the number of interactions that would be required for a dominant strategy to reach all individuals in the population. When this diameter was decreased, oscillations in fitness between different strategies, such as those shown in Section 3.15, were able to bring about the loss of diversity. This strong link between the structure of the interaction network and diversity was also demonstrated using populations of different size. The results of both of these experiments demonstrate that for cooperation to be maintained, populations must be large enough to accommodate the changes in abundance of cooperators and cheaters brought about by variation in fitness.

When populations were not able to maintain diversity, the experiments in Section 5.7, showed

that the introduction of novelty through mutation and migration were able to dampen the oscillations in fitness and allow diversity to be maintained. This occurred, because the introduction of novelty allowed populations to more quickly react to the changing targets of selection. For example, when a patch of resistant cells had formed after excluding the other cell types, selection would promote the invasion of the sensitive type, because it grows faster and is not susceptible to being killed by toxin in that patch.

# Chapter 6

# Conclusions

Theodosius Dobzhansky famously remarked that "nothing in biology makes sense except in the light of evolution" [36], signifying that we cannot fully understand the complexities of life without a grasp of the processes that brought them about. As has been demonstrated throughout this dissertation, however, evolution does not act alone as the sole driving force within spatially-structured populations. A vast array of ecological factors significantly affect evolutionary outcomes.

Chapter 3 demonstrated clearly that the fitness of cooperation can vary over time and space, depending on the complex interactions between ecological and evolutionary factors. Populations can only maintain cooperative behaviors when the combination of these factors permits cooperators to maintain abundances that enable them to stave off invasion from cheaters, which exploit the benefits of cooperation without contributing to it themselves. As these experiments confirmed, spatial structuring within populations can significantly aid in the evolution of cooperation. These three results form the basis for all of the work described in this dissertation.

Chapter 3 presented a number of experiments examining how environmental disturbance can bestow a selective advantage on cooperators. Positive niche construction through the production of surplus public good enabled cooperators and cheaters to survive in environments with increasing levels of disturbance. When cooperators were not able to first condition their environment, these severe levels of disturbance drove populations to extinction.

When disturbance was removed, cooperation no longer provided a benefit, and it was quickly purged from populations. However, once disturbance resumed, populations rapidly re-evolved cooperation and the protection that it provided. This result demonstrates that costly cooperation can readily evolve when it provides a sufficient fitness benefit. These results also indicate that battling cooperative behaviors such as antibiotic resistance may be substantially more difficult than simply suspending the use of one type of treatment, and may instead require long-term changes that continually repress selection for resistance.

Population bottlenecks, an extreme form of disturbance in which a large portion of the population is killed, have previously been reported to promote cooperation by reducing competition from cheaters [6]. Section 3.5, however, demonstrates that bottlenecks more generally act to amplify the differences in abundance between two or more strategies. Whether or not bottlenecks aid cooperation depends solely on the ability of the cooperator strain to maintain abundances equal to or greater in size than cheaters.

Natural populations are always limited by the availability of at least one resource [121]. Chapter 4 presented new insights into how the abundance of resource affects the evolution of cooperation garnered from experimental evolution studies and computational models targeting the cooperative formation of biofilms.

Although the data resulting from these experiments agree with those of a previous study [8], they offer a substantially different interpretation of the relationship between resource abundance and cooperation. Whereas previous work concluded that cooperation increased linearly as resource became more abundant, the experiments in Chapter 4 demonstrated the existence of a critical resource level, which corresponds to the point at which Hamilton's rule is satisfied, and cooperation is favored. Below this level, the relative costs of cooperation outweigh the benefits that it provides, and cooperation is not favored. Above this level, however, cooperation is beneficial, and will persist in the population. Interestingly, cooperation quickly plateaued after this transition point, indicating that cooperation was maintained at the lowest level that provided substantial benefit. Further investments in cooperation did not produce adequate increases in benefit.

After only approximately 45 generations, populations of *V. cholerae* that had evolved in resource-rich environments exhibited dramatic increases in cooperative biofilm production, while those populations that evolved in less rich environments did not. These results further demonstrate that even when all other conditions are equal, ecological factors, in this case resource abundance, can significantly alter evolutionary trajectories.

These experiments also demonstrated the power of investigating problems in ecology and evolution through the reciprocal use of computational models and bacterial populations. This multi-model approach enabled cooperative biofilm formation to be understood at a level that might not have been possible using just one system.

A common thread that connected all experiments presented in this dissertation was the spatial structuring of individuals within the population. Both Section 3.2 and numerous previous studies (e.g., [40,72,75]) have demonstrated that when interactions are localized, and individuals are more likely to interact with close relatives, diversity is more likely to be maintained. In the context of cooperation, a diverse population is one that can maintain both cheaters and cooperators. Understanding how the interactions among organisms affects diversity is critical to building a more complete picture of the forces that shape ecosystems. As such, this knowledge can inform conservation efforts and help us to understand the ramifications of living in an increasingly connected world. Diversity is also of critical importance in evolutionary computation, where premature convergence on one strategy often results in sub-optimal solutions [81]. While a number of diversity-preserving solutions have been proposed to combat this problem, e.g., [65], a deeper understanding of how the interactions that occur between candidate solutions in a population could greatly aid in maintaining diversity and finding more optimal solutions, through different selection strategies or crossover operations.

Using a model that represented interactions using arbitrary graphs, the experiments described in Chapter 5 investigated how the structuring of these groups affects the maintenance of diversity at a level of detail not possible with traditional lattice-based models. After investigating a variety of graph structures, the most informative indicator of a population's ability to maintain diversity

was found to be the diameter of its interactions, which represents the number of interactions that would be required for a dominant strategy to reach all individuals in the population. As Figure 3.15 showed in Chapter 3, fitness varies continuously over space and time, which can result in oscillations in the abundances of strategies. If the magnitude of these oscillations for a dominant strategy matches the diameter of interactions in the population, other strategies are excluded, and diversity is lost. Importantly, this loss of diversity occurs regardless of population size. Although the effects of population size have been the focus of much work, particularly in areas related to conservation biology (e.g., [134, 135]), this result offers a different perspective on how population size affects diversity.

Figure 3.15 also showed that when conditions change, and a behavior is once again favored by selection, that behavior can become more prevalent as long as it exists in the population. When diversity is lost, however, behaviors are purged from the population. No matter how favorable that behavior might be, it will not be able to proliferate unless it is re-introduced. Chapter 5 indicated that the continual introduction of novelty through mutation and migration allowed even small populations to main diversity. As this dissertation has shown, changing conditions cause fitness to change and selection to alternate between favoring and not favoring a behavior. Through migration and mutation, however, populations are able to react more quickly to such changes. This dampens the magnitude of oscillations in the abundance of strategies, and enables even small populations to maintain diversity.

Collectively, these studies have demonstrated that ecological effects, such as environmental disturbance, resource abundance, and social can have significant impacts on the evolution of cooperation. These ecological effects were shown to create variations in fitness over both time and space, providing selection with a continuously-moving target. By surviving through these variations in fitness, cooperation can evolve.

# Chapter 7

# Future Research Directions

The findings presented in this dissertation raise many other interesting avenues for continued research. This chapter outlines a few of them.

## 7.1 Prudent Cooperation through Quorum Sensing

This dissertation showed that selection for cooperation varies in response to both evolutionary and ecological processes. This creates a selective advantage for *facultative* cooperators, which choose to cooperate only when it is likely to be beneficial, over *obligate* cooperators like those modeled in this work. Although facultative cooperation can be achieved in an number of ways, such as through phenotypic noise [1], perhaps none are as prevalent in nature as *quorum sensing* (QS) [45]. Quorum sensing presents a large variety of questions that can have far-reaching impacts.

In 1970, Nealson, Platt, and Hastings [91] observed that prokaryotes could change their behaviors through the up- or downregulation of genes based on the concentration of signaling molecules in the surrounding environment. These signaling molecules, termed *autoinducers* (AIs), are continuously emitted by participating individuals. At low concentrations, AIs diffuse rapidly throughout the environment and elicit no response. However, the level of AI increases as the population grows. Once this level exceeds a threshold, receptors detect these signals and trigger changes in behavior. This includes activating the mechanisms that produce these AIs, resulting in an exponential

increase in the concentration of signal.

This quorum sensing process is now known to be an integral part of the lifestyles of bacteria and other microorganisms. Cooperative behaviors such as those that aid in nutrient uptake [117], provide protection against foreign invaders [12], and allow surface attachment [30] are among the wide variety of behaviors known to be controlled by QS. This list also includes biofilm formation [57, 126], which inspired the computational and microbial models used in this work. The integration of QS into these models to facilitate facultative cooperation could offer substantial new insights into both the evolution of cooperation and quorum sensing.

Quorum sensing is itself a form of cooperation, where individuals pay some cost to produce signals that benefit others by providing information about the conditions in the surrounding environment. This creates the potential for QS cheaters to exploit the signaling of others. Armed with the knowledge that QS is susceptible to exploitation, there has recently been great interest in the medical community to develop "anti-infective" treatments, which seek to disrupt the cooperative behaviors exhibited in infections through either the introduction of cheater mutants that do not send or receive signals, or by manipulating the signal itself [33, 34, 112]. It is hypothesized that treatments such as these exert less pressure to develop resistance than traditional antibiotic treatments, which target and kill infecting cells. By studying these types of techniques using models like those presented in this dissertation [4], a greater understanding of how populations respond to such treatments in a variety of environments can be gained and applied towards the development of effective treatments.

Microbes such as bacteria often live in complex, multi-species communities. Evolution may therefore favor signals that can only be interpreted by close relatives or the use of multiple signaling strategies. Although both inter- and intraspecific QS systems have been identified [70, 127], little is known about how diversification might arise in signal, receptor, and binding specificity, or how closely signal must match receptor in order to receive a signal. Recent computational models have shown that diversification is feasible when QS mediates a cooperative act, and that cheaters pave the way for the switch [42]; however, this two-signal, two-receptor system could neither address

the specific types of changes more likely to occur, nor whether changes in specificity would affect outcomes. The use of models that allow for a wide range of signals, receptors, and sensitivities could pave the way for a greater understanding of how diversity evolves in signaling.

## 7.2 Evolution's Effects on Ecology

Recently, Dobzhansky's famous quote was reappropriated by Grant and Grant, who wrote "Nothing in evolutionary biology makes sense except in the light of ecology" [48]. This, of course, only implies that ecology affects evolutionary biology. The more complete view, in which ecology is understood to affect evolutionary biology, and evolutionary biology is understood to affect ecology is beginning to gather support. With this perspective of a population's *eco-evolutionary dynamics*, where "Nothing in evolution or ecology makes sense except in the light of the other" [99], the behaviors of a population alter their environment, which in turn alters selection in that population.

The experiments described in this dissertation were designed to study the effects that ecology had on the evolution of cooperation. In each of the models used, however, ecology was also affected by evolution. As an example, the production of public goods, either in the form of biofilms or colicins, served as a form of niche construction [94], which made the environment more hospitable for cooperators. In natural populations, this more hospitable environment might increase the carrying capacity of the population or increase the potential number of interactions of each individual. Such effects were not explicitly studied; however, they could have dramatic impacts on the environment and on the evolution of cooperation. One potential study in this area could focus on how these niche-constructive behaviors can actually be cooperation's undoing.

## 7.3 Key Graph Properties and Dynamic Interaction Networks

Chapter 5 focused on how the network of interactions created by populations influenced their ability to maintain cooperation and diversity, finding the diameter to be a strong indicator. Otherwise, little is known about how the structure of these interaction networks affects evolution. This is a

major limitation, as lattice models, which are traditionally used in computational models, are not likely to accurately represent the interactions of natural organisms.

Aside from the fact that the use of graph structures for these purposes is in its infancy, the reason for this open area is likely because of the complexities inherent in graph structures. These complexities make examinations into the effects that changes to single parameters have difficult, because these changes often produce a cascade of other effects. For example, the experiments presented in Section 5.6 targeted the effects of population size. Changes in population size, however, produced changes in diameter, which was shown to be a key metric. Being able to alter population size without changing the diameter of the resulting network is an extremely difficult task, and even if it were accomplished, it is very likely that the resulting graph would differ in some other, potentially unknown, property.

Because of this, there is a considerable need for experiments informed by graph theory that identify the important properties of graphs that affect evolutionary outcomes and characterize their effects. Secondly, interactions within natural populations constantly change in response to growth, motility, death, and a variety of other factors. Therefore, the development of models that allow for such changes could permit studies of diversification, speciation, and communication in more realistic populations.

# Appendices

# Appendix A

# The SEEDS Platform for Evolutionary and Ecological Simulations

Evolutionary computation (EC) focuses on populations of individuals that are subjected to evolutionary processes such as mutation, selection, and inheritance [41]. Through the evolutionary process, individuals become better suited to their environments, which are defined by the user. Individuals can represent anything from candidate solutions to an optimization problem, such as equations fitting a dataset, to natural organisms that interact with each other and their environment. In any instance, evolution can be seen as a guided search process.

Digital evolution and other forms of evolutionary computation have been successfully used to address a number of fundamental questions in biology. (e.g., [72, 80]). In these areas, EC provides benefits that are not easily obtained, or simply impossible, when studying natural systems. Among these is the ability to study populations over thousands or even millions of generations, which is not feasible in most natural species. A second major benefit offered by EC is the complete transparency it provides to researchers, allowing them to observe each individual in the population, as well as its behaviors, interactions, and even its complete genetic encoding. These features offer researchers unique insights into the evolutionary trajectories of behaviors over evolutionary timescales, including how those behaviors evolve, whether they can be maintained, and the abun-

dances at which they exist in populations. Because of these benefits, there is great interest within the biological sciences to use EC techniques to generate and test hypotheses, supplement research using natural systems, and study phenomena that simply cannot be addressed otherwise.

Evolutionary computation has also been successfully used in engineering during the design and optimization processes (e.g, [63, 64]). In many instances, EC has produced novel and human-competitive results [74], cementing its status as a valuable tool throughout engineering.

Although several EC platforms are available, their widespread use is often limited by a number of factors. One of these is a sharp learning curve, which often presents a formidable barrier to those would-be users from biological fields who may lack significant computer programming experience. A second factor is the inflexibility of many platforms, which makes alterations or additions in functionality cumbersome. Because of these and other shortcomings, users often either create their own EC system, re-implementing standard functionality such as populations and selection, or avoid using EC altogether.

The SEEDS platform is intended to address these problems by providing a foundation upon which researchers can conduct experiments using a wide variety of techniques and models [21]. Because meeting the needs of all researchers would be an impossible task, SEEDS is designed as a modular system, where researchers can easily modify or extend its capabilities without also having to implement fundamentals. Through its well-defined interfaces and implementation in the Python programming language, users of all levels of programming experience are able to both use and extend SEEDS. Additionally, SEEDS places an emphasis on repeatability and data sharing, allowing users to easily recreate experiments and share their models.

To date, SEEDS has been used to explore a wide variety of topics related to evolution and ecology. SEEDS was initially developed in order to study the role spatial structure plays in maintaining diversity [23]. Additional studies have used SEEDS to explore the use and maintenance of horizontal gene transfer horizontal gene transfer [22], the evolution of public goods cooperation, the evolution of cooperation in probabilistic game theoretic models, and speciation and sexual dimorphism resulting from divergent selection in limited resource environments.

The SEEDS distribution includes several prebuilt components that enable experiments to be run immediately after installation. Among these is a model of the Rock-Paper-Scissors (RPS) game—**RPSCell**. Although this common game may seem trivial, it allows users of all ages and backgrounds to both experience and interact with many aspects that affect population dynamics. Such hands-on experimentation can be a powerful tool for learning fundamental processes in evolution and ecology. We will refer to the RPS example throughout the remainder this paper.

This chapter introduces the SEEDS platform as follows. Section A.1 presents the design of SEEDS and its organization. Section A.2 discusses how experiments are configured and performed. The plugin system and how it can be used to extend the functionality of SEEDS to match the research questions at hand is described in Section A.3. Finally, the significant additions planned for near-term inclusion are outlined in Section A.4.

## A.1 SEEDS's Design

In recent years, the Python programming language has experienced an enormous growth in its use in scientific computing. Part of this growth can be attributed to the fact that, in contrast to many scientific computing environments, Python is a general purpose programming language with a great deal of flexibility [101]. A second reason for Python's success is its broad user community, which has produced several powerful and easily-approachable packages for manipulating, analyzing, and displaying scientific datasets, such as NumPy [97], SciPy [68], and Matplotlib [66]. These and many other packages are easily obtained through Python's various package management utilities or Python distributions targeted specifically for use in science. Python's ubiquity is another major benefit, with versions available for most operating systems. Finally, Python is open source software, which means it does not impose a financial barrier to adoption. Although Python often does not always match lower-level languages such as C, C++, and Java in performance, especially in numerically-intense tasks, its ease of development and maintenance often more than make up for differences in performance on modern computing resources.

Figure A.1 shows the class diagram for SEEDS. A SEEDS Experiment consists of a Population of Cells. The possible phenotypes of a Cell are limited only by its implementation, which allows Cells to capture a wide variety of behaviors and genetic representations. The interactions among Cells are defined by the underlying Topology, an arbitrary graph structure which orients Cells in space. Cells also have access to Resources, which are distributed throughout space using a separate Topology. Finally, Actions allow the user to modify the environment, interact with cells, and produce output data, during the Experiment. Each of these is a Plugin, and users can easily modify and extend SEEDS through the creation of custom Plugins. In the following sections, we describe each of these classes in more detail.

### A.1.1 Experiment

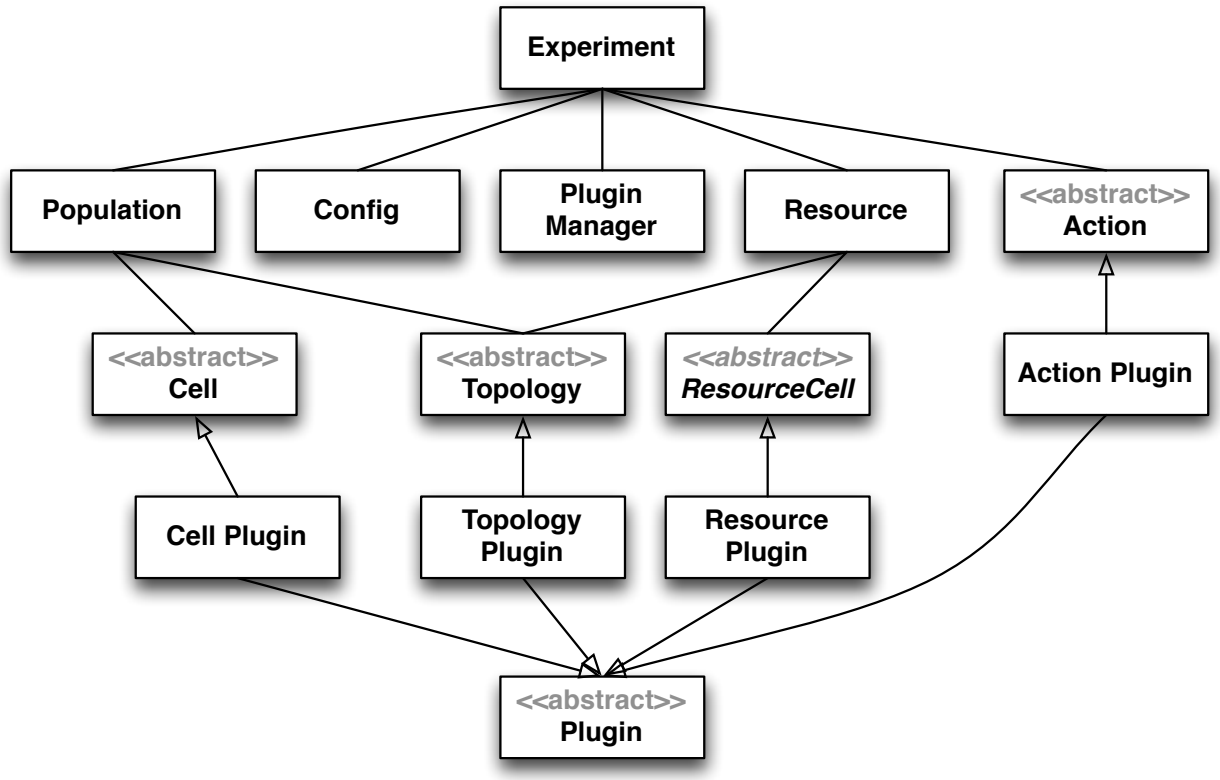



Figure A.1: SEEDS Class Diagram

The **Experiment** object controls all aspects of an experiment. This includes managing the

configuration, the population, and any available resources. During initialization, the Experiment object loads the specified configuration file and uses this information to create a population of individuals, environmental resources, and any actions which are scheduled to be performed, such as writing data or interacting with the population or resources, each of which is described below. During each unit of time, or *epoch*, the Experiment object updates the state of the population and resources, and executes any scheduled actions. Upon initialization, each Experiment is given a globally-unique identifier (GUID), allowing experiment data to be easily catalogued.

### A.1.2 Population

A population consists of a collection of individuals that reside on nodes in a graph structure. The graph defines the potential interactions of individuals. If the nodes in which two individuals reside are connected with by edge, those individuals have the potential to interact with each other, depending on the implementation of their Cell type. The use of arbitrary graphs to define the interactions within a population is one of the most powerful features of SEEDS. Experiments focus on one Population object; however, the interactions within that Population can be defined such that multiple independent subpopulations exist.

When a population is updated, a preconfigured number of individuals are chosen randomly with replacement, and the state of these individuals is updated based on the rules defined by the corresponding Cell instance. By default, the number of individuals updated is equal to the size of the population, so each individual is expected to be updated during each epoch, on average.

### A.1.3 Cell

In SEEDS, the **Cell** object is used to represent each individual, and is therefore often the primary focus of an Experiment. All Cells must define three methods: `__init__` (a constructor), `update`, and `teardown`. The constructor initializes all properties associated with that object, perhaps reading values from the configuration file. The `update` method is intended to update the state of that organism, and is executed during each epoch of the experiment, on average. The update method

may cause the Cell to reproduce, to consume some resource, to produce some resource, to interact with a neighboring Cell, to die, or to change state in response to any other event pertinent to the simulation. During each epoch, a subset of Cells from the population is chosen randomly and updated using this method. Finally, the `teardown` method is called whenever a Cell is removed from the population. This method is intended to handle any cleanup tasks necessary, such as the closing of files or the freeing of references. These methods are defined in the Cell base class. Any Cell object must be a subclass of this class, which also provides methods for retrieving and managing the neighbor list of each cell in the population.

A SEEDS cell is agnostic with respect to any particular evolutionary algorithm. What defines an individual is decided by the user: its representation, its behaviors and capabilities, its interactions, and ultimately its fitness. By creating a Cell subclass, users can implement a Cell class to suit their particular needs. For example, each cell could represent a state, such as alive or dead, as in Conway's Game of Life [46]. Alternately, each cell could encapsulate an artificial neural network, such as those evolved using NEAT [115]. A cell could even represent a program and have virtual hardware associated with it, as in the Avida system [95].

Continuing with the example, RPSCell is a cellular automaton model. Each Cell in the Population plays either the Rock, Paper, or Scissors strategy. When a Rock cell encounters a Paper cell, it becomes a Paper cell, because Paper covers Rock. However, if that Rock cell encountered a paper cell, it would remain a Rock, because Rock is not beaten by Scissors.

### A.1.4 Topology

The interactions of individuals within a population are defined by their **Topology**. In many forms of evolutionary computation, the interactions among individuals are defined by a lattice, where an individual residing in a node can interact with any of its 4 or 8 neighbors. To allow for more flexibility, SEEDS models interactions using arbitrary graphs. As with lattices, each individual occupies a node in the graph, and edges between two nodes represent a potential interaction between those corresponding individuals. The distribution of these edges, however, need not be regular as

in a lattice, but can instead define more complex interaction networks. This includes disconnected graphs, where certain subsets of the population do not interact with others.

Graphs are created and maintained using NetworkX [51], which provides a vast library of graph generators and allows for the creation of custom graph structures. Additionally, NetworkX offers functions that calculate many common graph metrics, which aid in understanding and comparing graphs.

Several topologies are included with the SEEDS distribution. As one example, the CartesianTopology topology is created by randomly placing nodes in a unit Cartesian plane and adding edges between nodes that are within a distance yielding an expected number of neighbors per node [23]. The CartesianTopology allows neighborhood sizes to grow linearly, instead of the geometric increase seen in lattice models as radius is increased.

Topologies are created by subclassing the *Topology* class, which provides a number of methods for maintaining the structure of the graph. Each topology must implement the `__init__` constructor method, which creates the appropriate graph, and the `teardown` method, which performs any necessary tasks before a topology is deleted. Topologies that support changes in structure during an experiment can additionally implement the `add_node`, `remove_node`, `add_edge`, and `remove_edge` methods, which handle the necessary changes to the graph.

### A.1.5 Config

The **Config** object manages the configuration for an Experiment. By wrapping Python's ConfigParser module, the Config object organizes the configuration into sections, one for each Experiment, Population, Cell, Resource, Topology, and Action. Each configuration section contains a set of parameter-value pairs, which define the value for a property in the respective object. These values can represent parameters for the object or define how that object behaves.

Typically, an Experiment will create a configuration from an input file. Cells such as RPSCell, for example, can then query the Config object to receive values to be used from the **RPSCell** section. RPSCell includes the *distance_dependent* parameter, which takes a boolean value specifying

whether or not closer neighboring Cells are more likely to interact than others. When querying the Config object, default values can also be specified for use when that parameter is unspecified in the configuration file.

The portion of an example configuration file that deals with RPSCell is shown in Listing A.1. This configuration will be expanded upon in Section A.2.3.

```
[RPSCell]
distance_dependent = True
```

Listing A.1: Segment of a configuration file that defines the behavior of RPSCells in a Population. Here, cells will be more likely to interact with nearby cells.

The Config object can also export an Experiment's current configuration. The resulting configuration file includes all parameters defined in the original configuration file, as well as all default values used and the seed for the random number generator. This greatly aids in reproducing experiments.

### A.1.6 Resource

Individuals can interact with each other and their environment through the consumption and production of resources. As shown in Figure A.2, each **Resource** specifies its own topology, which allows the distribution for each resource to be controlled. Each node in a resource contains a *ResourceCell* object, which defines the resource at that point in space. For example, SEEDS's *NormalResource* ResourceCell defines a resource as a level, an inflow, a decay, and a diffusion. The level defines the amount of resource currently present in that cell. Inflow defines the rate at which new resource enters that cell. Decay and diffusion result in resource being lost from the environment at a configured rate and resource flowing into neighboring ResourceCells at a configured rate, respectively.

The SEEDS distribution also contains the *SineResource* and *SquareResource* types, which vary the level of resource in a given cell according to a sine and square function, respectively. By subclassing the *ResourceCell* class, new resource types can easily be defined.

New resources must implement three common methods. The `__init__` constructor method,

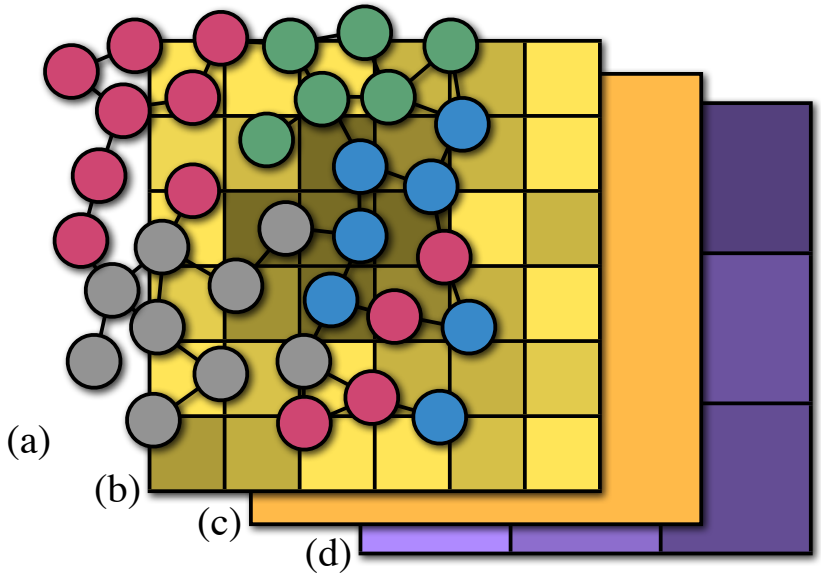



Figure A.2: Population and resource topologies. Each exists in the same unit Cartesian space; however, they partition that space differently. (a) The population topology is independent from the three resource topologies (b-d). The resource in (b) partitions the resource as a 6×6 lattice, while the resource in (c) is global, and resource in (d) partitions the resource topology as a 3×3 lattice.

which is executed when a ResourceCell is created, initializes the object and sets the initial level as configured. The `update` method adjusts the level of the resource according to the inflow, decay, and diffusion properties of that ResourceCell. Finally, the `teardown` method allows the ResourceCell to clean up its state before being deleted.

When a cell interacts with a resource, its coordinates in unit Cartesian space are projected onto the resource's topology. This interaction will then affect the level of the ResourceCell nearest to that coordinate. Therefore, the extent to which individuals share a particular resource pool depends on the partitioning of that resource's ResourceCells. A topology with fewer nodes will likely experience more overall competition for resource than a topology with more nodes. Although Resources use the same topology classes as populations, they are currently limited to lattices. This restriction will be lifted in a future release.

### A.1.7 Action

*Action*s define events that occur at specified times during an experiment. These actions perform tasks related to all areas of the experiment, from the population and its individuals to resources. Actions are also the primary way in which output data are produced. Some examples of actions included with SEEDS are listed in Table A.1.

Table A.1: Sample Actions Included with SEEDS

| Action | Description |
|---|---|
| PrintExperimentInformation | Print detailed information about the experiment and the software environment |
| PrintCellTypeCount | Print the abundances of each cell type in the population |
| SetResourceAvailability | Toggle the availability of a resource |
| StopOnConvergence | Stop the experiment when the number of different types of cells in the population falls below a threshold |

As an example, the *PrintCellTypeCount* action creates a comma-separated values (csv) file containing the number of each cell type present in the population at that time. These data can be used for further analysis after the experiment, or plotted, as demonstrated in Figure A.3. The *PrintCellLocations* action writes a csv file containing the location of each cell in the population, which can be used to visualize the distribution of strategies in space, as shown in Figure A.4.

All actions inherit from the *Action* class. Like most other SEEDS classes, actions must implement `__init__`, `update`, and `teardown` methods, which are executed when instances of that action are created, updated, and deleted, respectively. Each action also specifies an *epoch_start*, *epoch_end*, and *frequency*, which define when the action begins occurring, when it stops, and how frequently it occurs within that window of time, respectively. Plots similar to those shown in Figures A.3 and A.4 can be produced through the creation of Actions that plot population data during or after an Experiment. Several Actions that create plots are included in the `contrib` directory in the SEEDS distribution, including the *PlotCellTypeStack* and *DrawPopulation* actions used to create Figures A.3 and A.4, respectively.

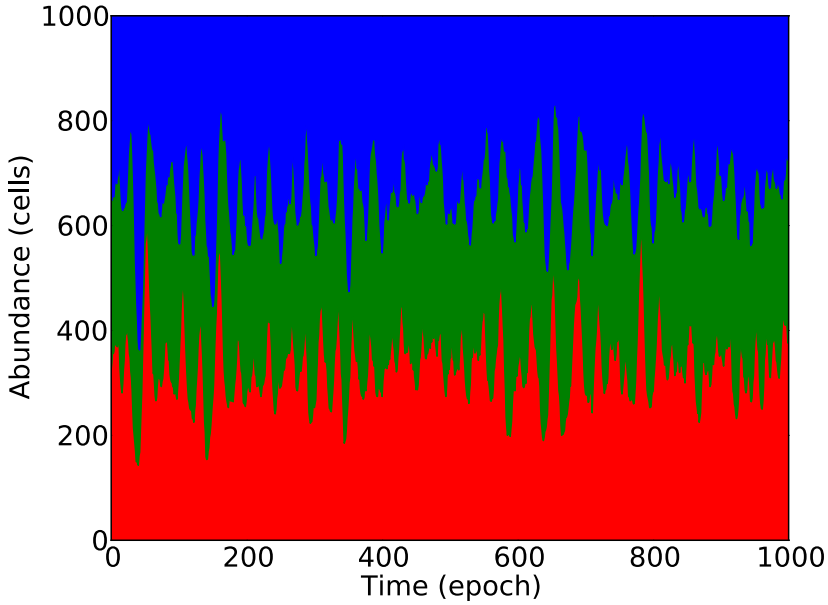



Figure A.3: Abundances of Rock (red), Paper (green), and Scissors (blue) cells over time in a population containing 1,000 RPSCell cells

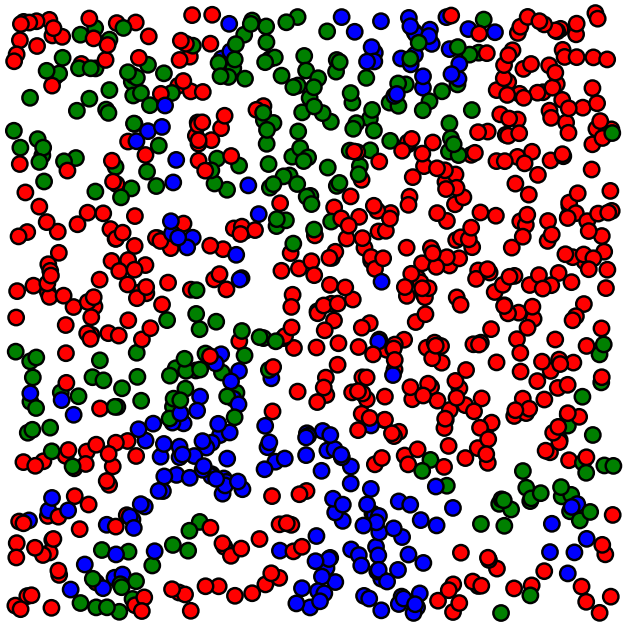


Figure A.4: Snapshot of a population of 1,000 Rock (red), Paper (green), and Scissors (blue) cells during an experiment using the CartesianTopology. In this example, each cell interacts with its 10 nearest neighbors, on average.

### A.1.8 Plugin

In SEEDS, each custom Cell, Topology, ResourceCell, and Action is also an instance of the **Plugin** class. These Plugins, described in detail in Section A.3, allow the functionality of SEEDS to be extended without recompilation or modification the SEEDS installation. Plugins also contain version information and other metadata, which allow for experiments to be replicated exactly.

### A.1.9 Plugin Manager

The **PluginManager** object receives and handles requests for Cell-, Topology-, Resource-, and Action- Plugins. The plugin manager scans the Plugins available in the directories specified by the configuration. If the plugin (and version) specified is available, an object of that type will be returned. Otherwise, an exception is raised.

## A.2 Using SEEDS

This section describes how SEEDS is typically used. First, Section A.2.1 discusses how SEEDS can be obtained and installed. Section A.2.2 details how experiments can be performed. Finally, Section A.2.3 introduces the SEEDS configuration file, which can be used to define a single experiment or family of experiments.

### A.2.1 Obtaining and Installing

SEEDS is open-source software, released under the Apache License 2.0[1], and is publicly available via a number of channels. The most straightforward way to install SEEDS is using the `pip` or `easy_install` tools. Alternately, all versions of SEEDS are available on the SEEDS development page[2], which also contains documentation, issue tracking, and development history. SEEDS can be installed from source using the standard Python Distribution Utilities.

1. http://www.apache.org/licenses/LICENSE-2.0
2. https://github.com/briandconnelly/seeds

SEEDS is designed to have minimal dependencies. A working installation will require Python 2.6.5 or greater (including 3.2) and NetworkX [51]. Although not required, SciPy [68], NumPy [97], and Matplotlib [66] are also recommended, due to their frequent use in third-party plugins for analysis and plotting.

### A.2.2 Running an Experiment

The most common way to conduct experiments is using the `runseeds.py` script. This script is installed with SEEDS, and includes a number of command-line options for configuring and running experiments.

Typically, a user will create a directory in which the experiment will be managed. That directory contains a configuration file, further described below, and an optional subdirectory containing any plugins to be used.

Experiments can also be run from within a Python interpreter or another script. Doing so simply requires creating an Experiment object and iterating over that object. A simple example of this is shown in Listing A.2. This flexibility allows experiments to easily be run in a number of ways, from command line tools and graphical user interfaces to web-based apps.

```
import seeds as S
experiment = S.Experiment('examples/Rock-Paper-Scissors/seeds.cfg
    ')

for epoch in experiment:
    print('Epoch {e} done'.format(e=epoch))
```

Listing A.2: Python code to create a SEEDS experiment, load the configuration file for the Rock-Paper-Scissors experiment, and perform the experiment, printing the status of the experiment at each epoch.

### A.2.3 Configuration

Configuration files define all aspects of an experiment, from the duration of the experiment, the Cell type and Topology to use, which resources are defined, which actions to run, and where to

place resulting data.

As shown in Listing A.3, a configuration file places each configurable item in its own section, indicated by brackets. In this example, the experiment, as defined in the **Experiment** section, will run for 1,000 epochs. It will use the *PrintCellTypeCount*, *PrintCellLocations*, and *StopOnConvergence* actions, each of which is configured below. It will find any third-party plugins in the *plugins* and *customcells* directories. Finally, any data written during this experiment will be placed in the *data* directory.

The population, defined in the **Population** section, will use the *RPSCell* cell type connected using the *CartesianTopology* topology. Here, the **:large** label appended to the topology indicates that the experiment will use the configuration specified in the **CartesianTopology:large** section, as opposed to the **CartesianTopology:small** section, which is also defined. This labeling allows a single configuration file to define multiple settings for each item, which simplifies managing the configuration of ensembles of experiments.

When this experiment is run, the *data* directory will be created. Any actions that produce data will write to this directory, such as *PrintCellTypeCount*, which writes a comma-separated-values (csv) file containing the abundance of each type of cell for a cellular automaton model, and *PrintCellLocations*, which prints a csv file listing the location of each cell.

Configuration files, including all default values and random number generator seeds necessary to replicate an experiment, are automatically created by SEEDS when run using the `runseeds.py` script with the `--genconfig` flag.

```
[Experiment]
epochs = 1000
actions = PrintCellTypeCount,PrintCellLocations,StopOnConvergence
plugin_dirs = plugins,customcells
data_dir = data

[Population]
cell = RPSCell
topology = CartesianTopology:large

[RPSCell]
distance_dependent = False

[CartesianTopology:small]
size = 2500
periodic = True
expected_neighbors = 10
remove_disconnected = False

[CartesianTopology:large]
size = 250000
periodic = True
expected_neighbors = 10
remove_disconnected = False

[PrintCellTypeCount]
start_epoch = 100
frequency = 1

[PrintCellLocations]
frequency = 100

[StopOnConvergence]
threshold = 3
```

Listing A.3: Example configuration for the *Rock-Paper-Scissors* experiment in which each cell plays either the Rock, Paper, or Scissors strategy against a randomly-chosen neighbor. This configuration file is available in the SEEDS distribution under `examples/Rock-Paper-Scissors`.

## A.3 Extending SEEDS

Although the SEEDS distribution includes a large number of Cells, Topologies, Actions, and Resources, it can be extended to address many other types of questions involving interactions within and among populations. Specifically, SEEDS has been designed to provide maximum customization and extensibility through a plugin system.

*Plugins* allow users to specify the behaviors, the environments, and the resources that best describe their model's needs through the creation of new Cells, Topologies, Resource Cells and Actions. By using plugins, users do not need to delve into SEEDS internals, modify the SEEDS installation, or wait for future releases in order to gain new capabilities. Instead, plugins are created locally and can be integrated into experiments immediately.

Plugins are created by subclassing the **Plugin** class. All plugin classes must define a number of properties that describe the plugin and allow experiments to specify specific versions to be used. These properties are listed in Table A.2. Additionally, plugins must implement the `__init__`, `update`, and `teardown` methods, which initialize the plugin, update its state, and perform any necessary cleanup activities, respectively.

Table A.2: Properties Defined by Each Plugin

| Property | Description |
|---|---|
| __name__ | The name of the plugin |
| __description__ | A detailed description of the plugin |
| __version__ | A tuple containing the major and minor version of this plugin |
| __author__ | The plugin author |
| __credits__ | Additional credits |
| __type__ | An integer specifying the type of the plugin (e.g., Cell, Topology, etc.) |
| __requirements__ | A list of required plugins and their version numbers |

All of SEEDS's built-in Cells, Topologies, Resource Cells, and Actions are implemented as plugins located in a system-wide plugin directory. These plugins can be modified or extended by creating local versions of them, allowing users to alter how they use SEEDS without affecting other users of the same SEEDS installation. The SEEDS distribution contains templates to aid in

the development of plugins of all types. A number of user-contributed plugins that are likely to be of use to a wider audience are included in the `contrib` directory of the SEEDS distribution.

The running Rock-Paper-Scissors is an example of a cellular automaton model. SEEDS Cells, however, can represent other types of models as well. In this example, we will construct a simple genetic algorithm model in which populations of individuals evolve to produce a target string. Here, the "genome" of each individual is an array of characters. At the beginning of the experiment, these genomes are initialized randomly using the 26-character Latin alphabet and the space character. The implementation of this cell is shown in Listing A.4. For brevity, comments and error checking are not included.

When a cell is updated, two random neighbors are chosen as parents with probability proportional to their fitness using roulette-wheel selection. A random, two-point crossover is then performed using their genomes, and mutations are applied site-by-site on the recombined genome. This design approximates a tournament selection with tournament size equal to the number of neighbors that each cell has.

## A.4 Future Directions

SEEDS is feature-rich, and has been used in several previous and ongoing research projects (e.g., [22, 23]). Nevertheless, a number of additions are planned that will increase the scale of experiments, foster repeatability of experiments, allow for easier distribution of plugins and configurations, offer new avenues for storing and interacting with resulting data sets, and make SEEDS a more approachable platform for use in education. This section details the significant enhancements planned for the near future.

### A.4.1 Experiments at Larger Scales

One potential shortcoming of experiments conducted using computational models such as SEEDS is their relatively-small population sizes. Typically, computational model populations consist of

```
class SentenceCell(Cell, Plugin):
    alphabet = string.ascii_lowercase + string.ascii_uppercase +
       ' '

    types = ['Sentence']

    def __init__(self,experiment,population,node,type=None,
                 name="SentenceCell",label=None):
        super(SentenceCell, self).__init__(experiment,population,
                                    node=node,type=0,name=name,
                                       label=label)
        self.target = self.experiment.config.get(self.
           config_section,'target')
        self.genome_length = len(self.target)
        self.mutation = self.experiment.config.getfloat(self.
           config_section,
                                 'mutation',default=0)
        self.genome = random.sample(self.alphabet,self.
           genome_length)
        self.calculate_fitness()

    def update(self):
        neighbor_fitnesses = [n.fitness for n in self.neighbors]

        parents = roulette_select(items=self.neighbors,
                                  fitnesses=neighbor_fitnesses,k=2)

        cp1 = random.randint(0,self.genome_length-1)
        cp2 = random.randint(0,self.genome_length-1)

        self.genome[:cp1] = parents[0].genome[:cp1]
        self.genome[cp1:cp2] = parents[1].genome[cp1:cp2]
        self.genome[cp2:] = parents[0].genome[cp2:]

        for i in range(self.genome_length):
            if random.random() < self.mutation:
                self.genome[i] = random.choice(self.alphabet)

        self.calculate_fitness()

    def calculate_fitness(self):
        """Fitness is 2^(number of matching characters)"""
        self.fitness = 1
        for i in range(self.genome_length):
            if self.genome[i] == self.target[i]:
                self.fitness *= 2
```

Listing A.4: SEEDS Cell implementing a genetic algorithm to evolve to match a target string. This cell's two parameters, *target* and *mutation*, can be defined in the **SentenceCell** section of a configuration file.

tens- to hundreds of thousands of individuals, which pales in comparison to microbial populations, which can be several orders of magnitude larger. Such smaller population sizes can affect evolutionary processes, and thereby alter the dynamics, ultimately leading populations to follow a entirely different trajectory.

A goal for future SEEDS development is to increase population sizes in order to more accurately model the dynamics of larger populations. Towards this objective, experiments have recently been performed with 1 million cells. Through improvements in design, increases in cyberinfrastructure, and the use of distributed processing, this scale can potentially be increased by an order of magnitude.

### A.4.2 Unit Testing Framework

One of the primary goals for SEEDS is to maintain a dependable platform that is backed by solid software engineering practices. Among these are unit tests, which are currently being integrated at all levels, from the core modules to plugins. By enabling the platform to be tested from experiment to experiment and version to version, we hope to maintain high levels of reliability and reproducibility.

### A.4.3 Self-Contained Experiments

SEEDS's flexible plugin system allows users to easily extend its functionality; however, this feature might also make sharing experiments that use many custom plugins and configurations more difficult. In order to facilitate sharing among users, a container format is being developed, which will permit users to include all files associated with a given experiment in a single file.

These container files will use a compressed, self-contained archive containing a manifest file describing the contents of the archive, one or more configuration files used to recreate an experiment, and all custom plugins used. The SEEDS distribution will include utilities for creating and editing these archives, as well as extracting individual elements from them.

### A.4.4 Flexible File IO

Although the comma-separated values (csv) files created by actions that write data are fairly standard and readable by a wide variety of software packages, greater flexibility in the output formats produced by SEEDS and its actions can offer further opportunities for interacting with experiment data sets. Although additional actions can easily be created that produce data in a different format, such duplication in effort is unnecessary. For example, by allowing users to write to a relational database, multiple sets of experiment data could be easily merged, queried, and analyzed, without the need to first transform or import the data.

Instead, SEEDS will integrate a modular layer for reading and writing data that allows plugins to be created for different data formats. Actions would then use the methods provided by this layer to read and write data, which would, in turn, use the data format specified by the user. This data layer will allow users to easily incorporate experiments into their own workflow.

### A.4.5 Graphical User Interface

Although SEEDS can be used to conduct research in a number of fields, it is also intended as a hands-on learning tool for use in the classroom. Through educational modules containing well-defined experiments, necessary configurations and plugins, as well as additional background information and prompts for exploration, SEEDS has the capability to be a powerful tool for students of all ages to observe and affect fundamental properties of evolution and ecology.

Currently, experiments can only be performed from the command line using the included `runseeds.py` script or through a Python interpreter. In order to reach a wider audience, an easy-to-use graphical user interface (GUI) will be developed to allow users to interact with experiments in a more intuitive way. Ideally, such a GUI would be built upon SEEDS's foundation of configurability and extensibility, allowing interfaces to nimbly accommodate features introduced in plugins. Although this extension will be a major undertaking, SEEDS's modularity promises to enable the development of first-class interfaces of all kinds.

# Bibliography

# BIBLIOGRAPHY


[1] M. Ackermann, B. Stecher, N. Freed, P. Songhet, W. Hardt, and M. Doebeli. Self-destructive cooperation mediated by phenotypic noise. *Nature*, 454(7207):987–990, 2008.

[2] R. Axelrod and W. Hamilton. The evolution of cooperation. *Science*, 211(4489):1390–1396, 1981.

[3] L. Barnett, E. Di Paolo, and S. Bullock. Spatially embedded random networks. *Physical Review E*, 76(5):56115, 2007.

[4] B. E. Beckmann, D. B. Knoester, B. D. Connelly, C. M. Waters, and P. K. McKinley. Evolution of resistance to quorum quenching in digital organisms. *Artificial Life*, 2012. In Press.

[5] J. Bonner. The origins of multicellularity. *Integrative Biology: Issues, News, and Reviews*, 1(1):27–36, 1998.

[6] M. Brockhurst. Population bottlenecks promote cooperation in bacterial biofilms. *PLoS One*, 2(7):e634, 2007.

[7] M. Brockhurst, A. Buckling, and A. Gardner. Cooperation peaks at intermediate disturbance. *Current Biology*, 17(9):761–765, 2007.

[8] M. Brockhurst, A. Buckling, D. Racey, and A. Gardner. Resource supply and the evolution of public-goods cooperation in bacteria. *BMC Biology*, 6(1):20, 2008.

[9] M. Brockhurst, M. Hochberg, T. Bell, and A. Buckling. Character displacement promotes cooperation in bacterial biofilms. *Current Biology*, 16(20):2030–2034, 2006.

[10] S. Brown, S. West, S. Diggle, and A. Griffin. Social evolution in micro-organisms and a Trojan horse approach to medical intervention strategies. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1533):3157–3168, 2009.

[11] C. Buckley and S. Bullock. Spatial embedding and complexity: The small-world is not enough. *Advances in Artificial Life*, pages 986–995, 2007.

[12] E. Cascales, S. Buchanan, D. Duche, C. Kleanthous, R. Lloubes, K. Postle, M. Riley, S. Slatin, and D. Cavard. Colicin biology. *Microbiology and Molecular Biology Reviews*, 71(1):158, 2007.

[13] H. Ceri, M. Olson, C. Stremick, R. Read, D. Morck, and A. Buret. The Calgary Biofilm Device: new technology for rapid determination of antibiotic susceptibilities of bacterial biofilms. *Journal of Clinical Microbiology*, 37(6):1771–1776, 1999.

[14] S. Chow, C. Wilke, C. Ofria, R. Lenski, and C. Adami. Adaptive radiation from resource competition in digital organisms. *Science*, 305(5680):84–86, 2004.

[15] A. Clatworthy, E. Pierson, and D. Hung. Targeting virulence: a new paradigm for antimicrobial therapy. *Nature chemical biology*, 3(9):541–548, 2007.

[16] B. Cohen. Incentives build robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72, 2003.

[17] J. Collins and C. Chow. Its a small world. *Nature*, 393(6684):409–410, 1998.

[18] J. Connell. Diversity in tropical rain forests and coral reefs. *Science*, 199(4335):1302–1310, 1978.

[19] B. D. Connelly, B. E. Beckmann, and P. K. McKinley. Resource abundance promotes the evolution of public goods cooperation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 143–150, 2010.

[20] B. D. Connelly, P. K. McKinley, and B. E. Beckmann. Evolving cooperative pheromone usage in digital organisms. In *Proceedings of IEEE Symposium on Artificial Life*, pages 184–191, 2009.

[21] B. D. Connelly, L. Zaman, and P. K. McKinley. The seeds platform for evolutionary and ecological simulations. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2012. To appear.

[22] B. D. Connelly, L. Zaman, P. K. McKinley, and C. Ofria. Modeling the evolutionary dynamics of plasmids in spatial populations. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 227–233, 2011.

[23] B. D. Connelly, L. Zaman, C. Ofria, and P. K. McKinley. Social structure and the maintenance of biodiversity. In *Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, pages 461–468, 2010.

[24] J. K. Conner and D. L. Hartl. *A primer of ecological genetics*. Sinauer Associates, 2004.

[25] J. Costerton, Z. Lewandowski, D. Caldwell, D. Korber, and H. Lappin-Scott. Microbial biofilms. *Annual Reviews in Microbiology*, 49(1):711–745, 1995.

[26] J. Costerton, P. Stewart, and E. Greenberg. Bacterial biofilms: a common cause of persistent infections. *Science*, 284(5418):1318–1322, 1999.

[27] T. Czárán, R. Hoekstra, and L. Pagie. Chemical warfare between microbes promotes biodiversity. *Proceedings of the National Academy of Sciences*, 99(2):786–790, 2002.

[28] T. L. Czárán and R. F. Hoekstra. Microbial communication, cooperation and cheating: Quorum sensing drives the evolution of cooperation in bacteria. *PLoS ONE*, 4(8):e6655, 08 2009.

[29] C. Darwin. *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*. Murray, London, 1859.

[30] M. Davey, N. Caiazza, and G. O'Toole. Rhamnolipid surfactant production affects biofilm architecture in Pseudomonas aeruginosa PAO1. *Journal of Bacteriology*, 185(3):1027–1036, 2003.

[31] M. Davey and G. O'toole. Microbial biofilms: from ecology to molecular genetics. *Microbiology and molecular biology reviews*, 64(4):847–867, 2000.

[32] R. Day, K. Laland, , and F. Odling-Smee. Rethinking adaptation: the niche-construction perspective. *Perspectives in Biology and Medicine*, 46(1):80–95, 2003.

[33] T. Defoirdt, N. Boon, and P. Bossier. Can Bacteria Evolve Resistance to Quorum Sensing Disruption? *PLoS Pathogens*, 6(7):430–435, 2010.

[34] T. Defoirdt, N. Boon, P. Sorgeloos, W. Verstraete, and P. Bossier. Quorum sensing and quorum quenching in Vibrio harveyi: lessons learned from in vivo work. *The ISME journal*, 2(1):19–26, 2007.

[35] S. Diggle, A. Griffin, G. Campbell, and S. West. Cooperation and conflict in quorum-sensing bacterial populations. *Nature*, 450(7168):411–414, 2007.

[36] T. Dobzhansky. Nothing in biology makes sense except in the light of evolution. *The American Biology Teacher*, 35(3):125–129, 1973.

[37] M. Doebeli and C. Hauert. Models of cooperation based on the Prisoner's Dilemma and the Snowdrift game. *Ecology Letters*, 8(7):748–766, 2005.

[38] M. Doebeli, C. Hauert, and T. Killingback. The evolutionary origin of cooperators and defectors. *Science*, 306:859–862, 2004.

[39] C. Dunford. Kin selection for ground squirrel alarm calls. *American Naturalist*, 111(980):782–785, 1977.

[40] R. Durrett and S. Levin. The importance of being discrete (and spatial). *Theoretical Population Biology*, 46(3):363–394, 1994.

[41] A. Eiben and J. Smith. *Introduction to evolutionary computing*. Springer Verlag, 2003.

[42] A. Eldar. Social conflict drives the evolutionary divergence of quorum sensing. *Proceedings of the National Academy of Sciences*, 108(33):13635–13640, 2011.

[43] K. Foster. Diminishing returns in social evolution: the not-so-tragic commons. *Journal of evolutionary biology*, 17(5):1058–1072, 2004.

[44] S. Frank. *Foundations of social evolution*. Princeton University Press, 1998.

[45] W. Fuqua, S. Winans, and E. Greenberg. Quorum sensing in bacteria: the LuxR-LuxI family of cell density-responsive transcriptional regulators. *Journal of Bacteriology*, 176(2):269–275, 1994.

[46] M. Gardner. Mathematical games: The fantastic combinations of John Conway's new solitaire game "Life". *Scientific American*, 223(4):120–123, 1970.

[47] J. Gore, H. Youk, and A. van Oudenaarden. Snowdrift game dynamics and facultative cheating in yeast. *Nature*, pages 253–256, 2009.

[48] P. Grant and B. Grant. *How and why species multiply: the radiation of Darwin's finches*. Princeton University Press, 2011.

[49] A. Griffin, S. West, and A. Buckling. Cooperation and competition in pathogenic bacteria. *Nature*, 430(7003):1024–1027, 2004.

[50] J. Grime. Competitive exclusion in herbaceous vegetation. *Nature*, 242:344–347, 1973.

[51] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference*, pages 11–15, Aug. 2008.

[52] A. R. Hall and N. Colegrave. How does resource supply affect evolutionary diversification? *Proceedings of the Royal Society B*, 274(1606):73–78, 2007.

[53] L. Hall-Stoodley, J. Costerton, and P. Stoodley. Bacterial biofilms: from the natural environment to infectious diseases. *Nature Reviews Microbiology*, 2(2):95–108, 2004.

[54] W. Hamilton. Selfish and spiteful behaviour in an evolutionary model. *Nature*, 228:1218–1220, 1970.

[55] W. D. Hamilton. The genetical evolution of social behaviour. I. *Journal of Theoretical Biology*, 7(1):1–16, 1964.

[56] W. D. Hamilton. The genetical evolution of social behaviour. II. *Journal of Theoretical Biology*, 7(1):17–52, 1964.

[57] B. Hammer and B. Bassler. Quorum sensing controls biofilm formation in Vibrio cholerae. *Molecular microbiology*, 50(1):101–104, 2003.

[58] A. Handel and M. Bennett. Surviving the bottleneck: transmission mutants and the evolution of microbial populations. *Genetics*, 180(4):2193–2200, 2008.

[59] I. Hanski. *Metapopulation ecology*. Oxford University Press, USA, 1999.

[60] G. Hardin. The tragedy of the commons. *Science*, 162(3859):1243–1248, 1968.

[61] A. Hastings and S. Harrison. Metapopulation dynamics and genetics. *Annual Review of Ecology and Systematics*, pages 167–188, 1994.

[62] C. Hauert and M. Doebeli. Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nature*, 428(6983):643–646, 2004.

[63] G. Hornby, J. Lohn, and D. Linden. Computer-automated evolution of an x-band antenna for NASA's space technology 5 mission. *Evolutionary Computation*, 19(1):1–23, 2011.

[64] J. Hu, E. Goodman, S. Li, and R. Rosenberg. Automated synthesis of mechanical vibration absorbers using genetic programming. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 22(3):207–217, 2008.

[65] J. Hu, E. Goodman, K. Seo, Z. Fan, and R. Rosenberg. The hierarchical fair competition (HFC) framework for sustainable evolutionary algorithms. *Evolutionary Computation*, 13(2):241–277, 2005.

[66] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, 9(3):90–95, May-Jun 2007.

[67] Y. Iwasa, M. Nakamaru, and S. Levin. Allelopathy of bacteria in a lattice population: competition between colicin-sensitive and colicin-producing strains. *Evolutionary Ecology*, 12(7):785–802, 1998.

[68] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001.

[69] G. Károlyi, Z. Neufeld, and I. Scheuring. Rock-scissors-paper game in a chaotic flow: The effect of dispersion on the cyclic competition of microorganisms. *Journal of theoretical biology*, 236(1):12–20, 2005.

[70] L. Keller and M. Surette. Communication in bacteria: an ecological and evolutionary perspective. *Nature Reviews Microbiology*, 4(4):249–258, 2006.

[71] B. Kerr. *Bacteriocins: ecology and evolution*. Springer, 2007.

[72] B. Kerr, M. Riley, M. Feldman, and B. Bohannan. Local dispersal promotes biodiversity in a real-life game of rock-paper-scissors. *Nature*, 418(6894):171–174, 2002.

[73] D. B. Knoester, P. K. McKinley, B. E. Beckmann, and C. Ofria. Directed evolution of communication and cooperation in digital organisms. In *Proceedings of the European Conference on Artificial Life (ECAL)*, pages 384–394, 2007.

[74] J. Koza. Human-competitive results produced by genetic programming. *Genetic Programming and Evolvable Machines*, 11(3):251–284, 2010.

[75] R. Kümmerli, A. Gardner, S. West, and A. Griffin. Limited dispersal, budding dispersal, and cooperation: an experimental study. *Evolution*, 63(4):939–949, 2009.

[76] K. Laland, F. Odling-Smee, and M. Feldman. Evolutionary consequences of niche construction and their implications for ecology. *Proceedings of the National Academy of Sciences*, 96(18):10242–10247, 1999.

[77] K. Lee and E. Ruby. Effect of the squid host on the abundance and distribution of symbiotic Vibrio fischeri in nature. *Applied and environmental microbiology*, 60(5):1565–1571, 1994.

[78] L. Lehmann. The evolution of trans-generational altruism: kin selection meets niche construction. *Journal of Evolutionary Biology*, 20(1):181–189, 2007.

[79] R. Lenski, M. Rose, S. Simpson, and S. Tadler. Long-term experimental evolution in Escherichia coli. I. adaptation and divergence during 2,000 generations. *American Naturalist*, 138(6):1315–1341, 1991.

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

[81] Y. Leung, Y. Gao, and Z. Xu. Degree of population diversity-a perspective on premature convergence in genetic algorithms and its markov chain analysis. *Neural Networks, IEEE Transactions on*, 8(5):1165–1176, 1997.

[82] T. Mah and G. O'Toole. Mechanisms of biofilm resistance to antimicrobial agents. *Trends in microbiology*, 9(1):34–39, 2001.

[83] P. Marsh. Dental plaque as a microbial biofilm. *Caries research*, 38(3):204–211, 2004.

[84] E. Mayr. *Systematics and the origin of species, from the viewpoint of a zoologist*. Harvard University Press, 1942.

[85] R. McElreath and R. Boyd. *Mathematical models of social evolution: A guide for the perplexed*. University of Chicago Press, 2007.

[86] J. H. Miller. *Experiments in molecular genetics*. Cold Spring Harbor Laboratory, 1972.

[87] J. Mitteldorf and D. Wilson. Population viscosity and the evolution of altruism. *Journal of Theoretical Biology*, 204(4):481–496, 2000.

[88] M. Mobilia. Oscillatory dynamics in rock-paper-scissors games with mutations. *Journal of Theoretical Biology*, 264(1):1–10, 2010.

[89] C. Nadell, B. Bassler, and S. Levin. Observing bacteria through the lens of social evolution. *Journal of Biology*, 7(7):27, 2008.

[90] J. Nahum, B. Harding, and B. Kerr. Evolution of restraint in a structured rock–paper–scissors community. *Proceedings of the National Academy of Sciences*, 108(Supplement 2):10831–10838, 2011.

[91] K. Nealson, T. Platt, and J. Hastings. Cellular control of the synthesis and activity of the bacterial luminescent system. *Journal of Bacteriology*, 104(1):313–322, 1970.

[92] M. Nowak and R. May. Evolutionary games and spatial chaos. *Nature*, 359(6398):826–829, 1992.

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

[94] F. Odling-Smee, K. Laland, and M. Feldman. *Niche construction: the neglected process in evolution*, volume 37. Princeton Univ Pr, 2003.

[95] C. Ofria and C. Wilke. Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229, 2004.

[96] H. Ohtsuki, C. Hauert, E. Lieberman, and M. Nowak. A simple rule for the evolution of cooperation on graphs and social networks. *Nature*, 441(7092):502–505, 2006.

[97] T. E. Oliphant. *A Guide to NumPy*, volume 1. Trelgol Publishing, 2006.

[98] G. O'Toole, L. Pratt, P. Watnick, D. Newman, V. Weaver, and R. Kolter. Genetic approaches to study of biofilms. *Methods in enzymology*, 310:91–109, 1999.

[99] F. Pelletier, D. Garant, and A. Hendry. Eco-evolutionary dynamics. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1523):1483–1489, 2009.

[100] J. W. Pepper and B. B. Smuts. A mechanism for the evolution of altruism among non-kin: Positive assortment through environmental feedback. *American Naturalist*, 160:205–213, 2002.

[101] F. Pérez, B. E. Granger, and J. D. Hunter. Python: An ecosystem for scientific computing. *Computing in Science and Engineering*, 13:13–21, 2011.

[102] E. Pestelacci and M. Tomassini. Hawks and doves in an artificial dynamically structured society. In *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 466–473, 2008.

[103] T. Pfeiffer, S. Schuster, and S. Bonhoeffer. Cooperation and competition in the evolution of ATP-producing pathways. *Science*, 292(5516):504–507, 2001.

[104] T. G. Platt and J. D. Bever. Kin competition and the evolution of cooperation. *Trends in Ecology & Evolution*, 24(7):370–377, 2009.

[105] F. Prado and B. Kerr. The evolution of restraint in bacterial biofilms under nontransitive competition. *Evolution*, 62(3):538–548, 2008.

[106] G. Price et al. Selection and covariance. *Nature*, 227:520–521, 1970.

[107] D. Queller. Genetic relatedness in viscous populations. *Evolutionary Ecology*, 8(1):70–73, 1994.

[108] D. Queller. Expanded social fitness and Hamilton's rule for kin, kith, and kind. *Proceedings of the National Academy of Sciences*, 108(Supplement 2):10792–10799, 2011.

[109] D. Queller, E. Ponte, S. Bozzaro, and J. Strassmann. Single-gene greenbeard effects in the social amoeba dictyostelium discoideum. *Science*, 299(5603):105–106, 2003.

[110] P. B. Rainey and K. Rainey. Evolution of cooperation and conflict in experimental bacterial populations. *Nature*, 425:72–74, 2003.

[111] D. Rankin, K. Bargum, and H. Kokko. The tragedy of the commons in evolutionary biology. *Trends in Ecology & Evolution*, 22(12):643–651, 2007.

[112] K. P. Rumbaugh, S. P. Diggle, C. M. Watters, A. Ross-Gillespie, A. S. Griffin, and S. A. West. Quorum sensing and the social evolution of bacterial virulence. *Current Biology*, 19(4):341–345, 2009.

[113] J. Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.

[114] J. Smith and G. Price. The logic of animal conflict. *Nature*, 246(5427):15–18, 1973.

[115] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[116] P. S. Stewart and J. W. Costerton. Antibiotic resistance of bacteria in biofilms. *The Lancet*, 358(9276):135–138, 2001.

[117] A. Stintzi, K. Evans, J. Meyer, and K. Poole. Quorum-sensing and siderophore biosynthesis in Pseudomonas aeruginosa: lasR/lasI mutants exhibit reduced pyoverdine biosynthesis. *FEMS Microbiology Letters*, 166(2):341–345, 1998.

[118] J. E. Strassmann, O. M. Gilbert, and D. C. Queller. Kin discrimination and cooperation in microbes. *Annual Review of Microbiology*, 65:349–367, 2011.

[119] X. Tan, D. Kim, N. Usher, D. Laboy, J. Jackson, A. Kapetanovic, J. Rapai, B. Sabadus, and X. Zhou. An autonomous robotic fish for mobile sensing. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5424–5429. IEEE, 2007.

[120] K. Thelin and R. Taylor. Toxin-coregulated pilus, but not mannose-sensitive hemagglutinin, is required for colonization by Vibrio cholerae O1 El Tor biotype and O139 strains. *Infection and immunity*, 64(7):2853–2856, 1996.

[121] D. Tilman. *Resource competition and community structure*. Princeton University Press, 1982.

[122] D. Tilman and P. Kareiva, editors. *Spatial ecology: the role of space in population dynamics and interspecific interactions*. Princeton University Press, 1997.

[123] V. Van Noort, B. Snel, and M. Huynen. The yeast coexpression network has a small-world, scale-free architecture and can be explained by a simple model. *EMBO reports*, 5(3):280–284, 2004.

[124] K. Visick and M. McFall-Ngai. An exclusive contract: specificity in the Vibrio fischeri-Euprymna scolopes partnership. *Journal of Bacteriology*, 182(7):1779–1787, 2000.

[125] J. Von Neumann. *Theory of self-reproducing automata*. University of Illinois Press, 1966.

[126] C. Waters, W. Lu, J. Rabinowitz, and B. Bassler. Quorum sensing controls biofilm formation in Vibrio cholerae through modulation of cyclic di-GMP levels and repression of vpsT. *Journal of bacteriology*, 190(7):2527–2536, 2008.

[127] C. M. Waters and B. L. Bassler. Quorum sensing: cell-to-cell communication in bacteria. *Annual Review of Cell and Developmental Biology*, 21:319–346, 2005.

[128] D. Watts and S. Strogatz. Collective dynamics of "small-world" networks. *Nature*, 393(6684):440–442, 1998.

[129] S. West, S. Diggle, A. Buckling, A. Gardner, and A. Griffin. The social lives of microbes. *Annu. Rev. Ecol. Evol. Syst.*, 38:53–77, 2007.

[130] S. A. West, A. S. Griffin, and A. Gardner. Social semantics: altruism, cooperation, mutualism, strong reciprocity and group selection. *Journal of Evolutionary Biology*, 20:415–432, 2007.

[131] W. Whewell. *The philosophy of the inductive sciences: founded upon their history*. Frank Cass and Company, 1840.

[132] D. Wilson. Social semantics: toward a genuine pluralism in the study of social behaviour. *Journal of Evolutionary Biology*, 21(1):368–373, 2008.

[133] S. Wright. Coefficients of inbreeding and relationship. *American Naturalist*, 56(645):330–338, 1922.

[134] S. Wright. Evolution in mendelian populations. *Genetics*, 16(2):97–159, 1931.

[135] S. Wright. Size of population and breeding structure in relation to evolution. *Science*, 87(2263):430–2264, 1938.

[136] V. Wynne-Edwards. *Evolution through group selection*. Blackwell Scientific Publications, 1986.

[137] J. B. Xavier, E. Martinez-Garcia, and K. R. Foster. Social evolution of spatial patterns in bacterial biofilms: When conflict drives disorder. *The American Naturalist*, 174(1):1–12, 2009.

[138] F. Yildiz, X. Liu, A. Heydorn, and G. Schoolnik. Molecular analysis of rugosity in a Vibrio cholerae o1 El Tor phase variant. *Molecular microbiology*, 53(2):497–515, 2004.