Evolution of Finite Populations in Dynamic Environments

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Liekens, Anthony M.L.

Evolution of Finite Populations in Dynamic Environments / by Anthony M.L. Liekens. - Eindhoven : Technische Universiteit Eindhoven, 2005.
Proefschrift. - ISBN 90-386-2637-1
NUR 922
Subject headings: co-evolution / dynamic environments / evolutionary game theory / finite population size / genetic algorithms / genetic drift / Markov models / population genetics

The work in this thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithms), IPA dissertation series 2005-11

Front cover image based on “Predestination” (1951) by M.C. Escher.

Back cover image represents a simulation of the evolution of a population of $r = 19$ individuals with mutation rate $\mu = 0.04$, according to the evolutionary model of Rock-Paper-Scissors as laid out in section 6.7 of this text.

Printed by Printservice Technische Universiteit Eindhoven, Eindhoven, the Netherlands

Evolution of Finite Populations in Dynamic Environments

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Eindhoven,
op gezag van de Rector Magnificus, prof.dr.ir. C.J. van Duijn
voor een commissie aangewezen door het College voor Promoties
in het openbaar te verdedigen
op dinsdag 5 juli 2005 om 16.00 uur

door

Anthony Mariette Louis Liekens

geboren te Sint-Niklaas, België

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr. P.A.J. Hilbers
en
prof.dr. E.H.L. Aarts

Copromotor:
dr.ir. H.M.M. ten Eikelder

Contents

Chapter 1

Introduction

In this work, we are interested in the long term evolutionary behavior of small populations of haploid and diploid individuals in dynamic fitness environments. In this chapter, we introduce these concepts and give an outline of the text.

Small populations. This work is concerned with the mathematical modeling of evolutionary dynamics in small populations. In biological evolution, fitter individuals are selected from a population to produce a new population of offspring at repetitive generations through the processes of inheritance and variation of traits. Evolution is a stochastic process, since reproduction of individuals contains random elements. The influence of random variation or fluctuations of genes is much larger in small populations than in large populations.

As an illustration of such fluctuations, consider coin tosses which, on average, produce heads and tails equally. It is however unlikely to produce heads and tails in an equal number when a coin is tossed only a few times. As more coins are tossed, the expected result is closer to this equal ratio of heads and tails. There is a probability of one in eighteen that a series of 10 coin tosses turns out to have 80% or more heads. The chance of throwing 80% or more heads in a series of 100 coin tosses is only about one in 1.8 billion.

Such sampling effects are also of importance in small breeding populations. Assume a parent population which has a given proportion p of a specific allele for a given gene, and alleles are passed on to the next generation without any interference of natural selection. In similarity with the coin tossing example, there is a higher probability of ending up with a child population that has a proportion p' that is further away from p if this offspring population is small. Since the offspring population is a random sample of the parent population, a small population is subject to larger fluctuations of allele frequencies. If the frequencies of alleles in the offspring population diverge from the parent population, the population is said to have genetically drifted away from the initial frequencies.

As the proportion of an allele in a small population drifts up and down over the course of successive generations, the population may eventually become fixated, i.e., the proportion of the allele in the population ends up in 0, as the allele disappears from the population, or 1, as the allele becomes the only allele in the population. For smaller populations, this effect of fixation occurs, on average, earlier than in larger populations. Only through a mutation, the population can escape from this state of fixation.

Small populations may thus introduce frequencies of alleles that seem at odds with natural selection. In small populations, the evolutionary dynamics of natural selection and drift are both at work. In this text, we study the dynamics of small populations in a set of differing environments that determine the forces of natural selection. For each of the specific environments, we study how the evolutionary forces of selection and drift influence the observed dynamics.

Dynamic environments. The focus of this thesis is the dynamics of small populations in environments that change over time.

Commonly, mathematical models to study the evolutionary dynamics of a population assume an environment that does not change over time. In such environments the absolute fitness – the capability to produce offspring – of a specific genotype is kept fixed over time as the population evolves. In evolutionary systems with such static fitness environments, the population of individuals initially explores mutant genotypes to find those genotypes that generate a high fitness. A high mutation rate in the reproductive process of the individuals allows for exploration of the genotype space, as new mutant genotypes are frequently introduced into the population. Once the population has genotypes with high fitness, the population may exploit these advantageous alleles. A population is said to converge, or become fixated for a genotype if the proportion of the genotype increases in the population and no further noticeable exploration of mutant genotypes takes place. A low mutation rate allows the population to better converge to a certain genotype. Note that the genotype to which the population converges is not necessarily the optimal genotype.

In contrast with such environments that assume a fixed absolute fitness for a genotype over time, there are numerous examples of environments in nature that change over time, and have their effect on the fitness of individuals of a certain genotype. Differing weather patterns, or sudden shortages of resources may have their temporary effects on the fitness of genotypes. Because of such environmental changes, a deleterious genotype may become advantageous, and vice versa.

Similarly, an individual's fitness may be dependent on the other individuals of its population. As an example, consider environments that assume fitness sharing, i.e., individuals in the population with the same genotype experience a fitness penalty. As the population evolves over time and the composition of the population thus changes over time, the fitness of a specific genotype may fall or rise as more or less individuals of that genotype are present in the population.

The fitness of a genotype in one population may also be dependent on another evolving population, i.e., in the co-evolution of two coupled populations. As an example, consider the capability of running quickly in populations of foxes and hares. If the population of foxes evolves a way to run faster, the fitness of hares decreases as a result of the evolutionary progress of the foxes. The fitnesses of foxes and hares are dependent on the evolutionary dynamics of the opponent population and as a result, the absolute fitnesses of genotypes in co-evolution change over time.

In dynamic fitness environments, the emphasis is more on the continuing exploration of the genotype space, rather than on exploitation of genotypes with high fitness, since the fitnesses of individuals constantly change. In contrast with populations in static environments, these populations require mechanisms to better track changes in the environment, and constantly explore the genotype space for genotypes with high fitness. The populations may not be able to converge to optimal genotypes in the long run as was the case with static fitness environments, but may, even in the long run, require a high amount of exploration, and thus a higher rate of mutation in order to perform optimally. In this thesis, we study how small populations behave in the long run as their fitness is subject to continuous change.

Models of infinite populations are commonly used to study the evolutionary dynamics of populations. For fitness functions that remain static over time, Markov models of finite populations have been successfully adopted to study the influence of drift on the predictions of infinite models. In this thesis, we adopt an experimental mathematics approach with Markov models to study the influence of finite populations and drift in environments that change over time.

Focus disciplines. The contents of this thesis relates to three closely related scientific disciplines.

Firstly, population genetics studies the evolutionary dynamics of biological populations. The relation to the contents of this thesis is self-explanatory. Secondly, evolutionary game theory is concerned with the dynamics of populations evolving strategies for games. In these evolving populations, individuals play simple games against other members of the population in order to gather fitness. Individuals that adopt strategies with high payoffs for the game, and thus a high fitness, can spread in the population. As a direct result of this construction, the fitness of an individual is dependent on the composition of the population, and is consequently dependent on the evolutionary dynamics of the population. We study the behavior of small populations for a set of such small games. Thirdly, genetic algorithms are computer simulations of evolving populations, that can be adopted to find approximate solutions for optimization problems. Recently, optimization problems that change over time, and implementations of co-evolution have gained interest in the applications of the search technique. The theoretical study of models of genetic algorithms may give insights in how to design and develop good optimizers and are consequently also of interest to this text.

In evolutionary game theory and genetic algorithms, it is common practice to study the dynamics of populations with haploid individuals. In population genetics, which studies the evolution of higher order species, diploid populations are commonly assumed. At the construction of our models, we consider both haploid and diploid reproduction cycles such that we can compare their expected performances in small populations, and in dynamic fitness environments.

In this thesis, the emphasis is on predictions that relate to population genetics.

However, since evolutionary game theory and genetic algorithms are also based on models of evolution, the predictions for the different disciplines are thus also relevant to each other. The models and notation adopted are based on those commonly used in genetic algorithms. One model thus serves as a means to study the evolutionary dynamics for the three focus disciplines. It should be noted that, in the text, the jargon of the different disciplines comes together, and specific concepts borrowed from one discipline often relate to a similar concept of another discipline. E.g., a *phenotype space* of population genetics relates to a *search space* of an *optimization problem* of a genetic algorithm, etc. At the mathematical level of abstraction in this text, the distinction between the different models vanishes.

Thesis outline. Chapter 2 gives an overview of models that assume infinite and finite population size. We review the relevant literature from our three focus disciplines and give an overview of the methodology applied in this thesis. In Chapter 3, we construct finite and infinite models of haploid and diploid populations in static fitness environments. In Chapter 4, we give some examples of how the finite population models behave in static fitness environments, to build up a basic understanding of how finite populations behave in the absence of a dynamical fitness environment. In Chapter 5, we develop models and analyze the behavior of systems where the dynamics of the fitness function is defined explicitly. Chapter 6 studies finite population models that are subject to frequency dependent selection, defined in similarity to the models commonly adopted in evolutionary game theory. In Chapter 7, we construct and study models representing the co-evolution of finite population models. Chapter 8 summarizes and discusses the conclusions of this thesis.

Chapter 2

Infinite and Finite Populations

Many models that are used to study evolving systems, assume infinitely large populations. Biological populations are in many cases indeed very large, such that the approximation by infinitely large populations often offers valid predictions of the behavior in very large populations. As a surplus, the assumption of infinite populations simplifies the mathematical modeling. There are, however, many cases where populations are small, and where the approximation of infinitely large populations is no longer valid. In these small populations, the stochastic sampling in each generation plays an important role. Random sampling of individuals leads to stochastic fluctuations in frequencies of alleles within such populations. This effect is called genetic drift. The effect of genetic drift is more evident in smaller populations, and not existent in infinitely large populations.

This chapter introduces the effects of finite population size in comparison to models of infinite population size. We give an overview of existing literature with respect to such models in population genetics, evolutionary game theory and the study of evolutionary algorithms. We also introduce the experimental mathematics methodology adopted in this thesis.

2.1 Models of Populations

2.1.1 Infinite populations

Populations are collections of organisms of a particular species, possibly in a specific geographical location. Commonly, biological populations whose dynamics are studied are generally large. The world population is currently estimated to be around 6.4 billion people, the population of doves in Northern America is estimated to be around 475 million, and many penguin species' populations range from 100 thousands to several millions, to give just a few arbitrary statistics.

When parents give birth to children, and if the chances of the sexes of these children are 50% males and 50% females, then we also expect that, within very large populations of descendants, the expected ratio among these children's sexes is very close to one. This is according to the law of large numbers, which expresses that in a series of independent equal trials with the same probability p of success (e.g., p is the probability of being female) in each trial, the chance that the proportion of successes deviates from p converges to zero as the number of trials increases and goes to infinity, see e.g., Feller (1968). In other words, as populations' sizes increase, the effects of random sampling in these populations vanish.

Similarly, we can model a very large (haploid) population where 50% of the population has a certain hereditary characteristic. Assuming there is no selective pressure on the given trait, parents are uniformly selected from this population. If paired parents have this characteristic, it is inherited by their child, and if only one of the parents has the characteristic, then we assume that it is inherited by one out of two of their children. Consecutive generations of very large populations maintain the initial ratio of 50% : 50%.

The stochastic effects of random sampling are faint enough that they can be ignored by assuming an infinite population size. Consequently, if models are to be built to study the dynamics of very large populations, an infinite population size can justifiably be assumed, ignoring the faint stochastic effects of sampling. As a surplus, the assumption of infinitely large populations simplifies the mathematics of the models, as random sampling and the stochasticity it produces can be disregarded.

2.1.2 Finite population models

Models with infinite population size predict expectations of frequencies of alleles, or representative populations according to the deterministic dynamics of the system, where no random sampling is present. A model that assumes a finite population size is stochastic by nature, since a population is sampled at each generation. For a stochastic model of finite populations, the expectation of frequencies of alleles in the population in the next generation is equal to the frequencies of the corresponding deterministic infinite model. By implying a finite population size, the dynamics of variance of this expectation should also be considered. As populations are smaller, the stochastic, variational effects of repetitively sampling consecutive populations may overwhelm the predicted expectations of the infinite population model.

As in the previous section, consider a small (haploid) population where 50% of the individuals have a certain heritable characteristic on which no selective pressure is exerted, and where children inherit one of the genes that determine their parents' characteristics for that trait. As generations are constructed repeatedly by combining random parents, fluctuations in allele frequencies appear, because of the stochastic nature of sampling a new small population at each generation. Populations diverge, or drift away from the initial frequencies, until they become fixated in either one of the alleles. A population is said to be fixated if all of the individuals in the population are of the same type. This is in contrast with an infinite population model, which would remain at a constant proportion of 50% for either allele. This effect of dropping the assumption of infinitely large populations is known as genetic drift, and was first introduced by Wright (1931). Genetic drift is the stochastic effect of random sampling in finite populations, which causes frequencies of genes to diverge from the expected deterministic dynamics of infinite populations.

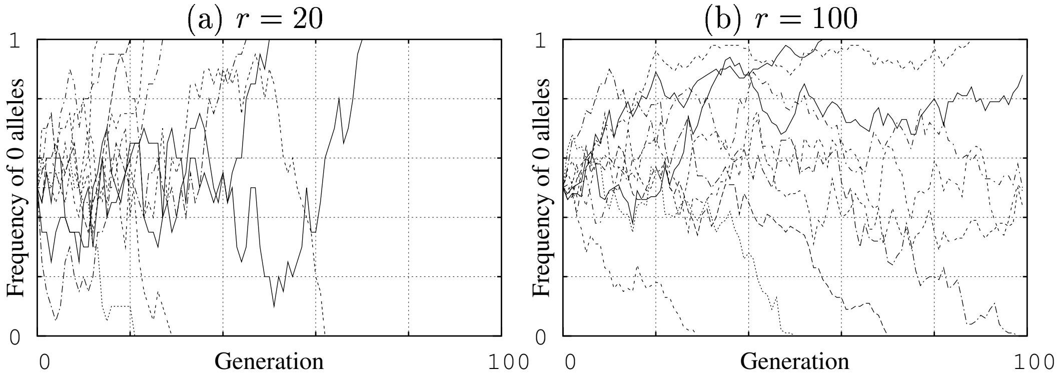


Figure 2.1: The effect of population size r on the rate of divergence and the time to fixation of populations, with no selective or variational pressure. Multiple simulations with the same population size were started with a ratio of $1:1$ alleles, and are plotted in each figure.

Figure 2.1 shows simulations of this example with varying population sizes r. As r increases, the rate of divergence becomes more slowly, and the time to fixation becomes longer. However, no matter how big the finite population size, fixation will occur, if no evolutionary forces – such as selective or mutational pressure – are present. Genetic variability is lost randomly over time because of drift. Wright (1931) has shown that the rate of loss in variation is $1/r$ at each generation. Kimura & Otha (1969) elaborately study the time to fixation using diffusion equations, with the time to fixation of a locus with two alleles being approximately $1.4r$, if the initial generation has 50% of either allele.

Genetic drift may have an important impact on the evolutionary history of a population. In a population bottleneck, where a population is suddenly reduced in size, dramatical changes in the allele frequency of the population may be introduced. Many of the advantageous alleles may become removed from the population, because of an “unlucky” sample of alleles at the time of the population bottleneck.

A similar effect is known as the founder’s effect, and occurs when a small population of individuals migrates to start a new population. Only a small sample of the alleles of the original population may be present in the new population. In extreme cases, founding populations may give rise to speciation and the evolution of a new species. More commonly, the founder’s effect may introduce genetic diseases. As an example, the Amish populations in the United States have grown from a small migrating group, tend to marry within this group, and exhibit polydactyly (extra fingers and toes) more commonly than the United States population at large. It is assumed that the initial population of Amish settlers had a high proportion of individuals with this genetic pathology, allowing it to be more prominent in the current population.

Genetic drift has become an important part of the current understanding of evolution, and has led to the neutral theory of evolution, or the neutral theory of molecular evolution, introduced by Kimura (1968) in the 1960s and 1970s and given full account in (Kimura 1986; Kimura 1994). The theory has become complementary to Darwinian evolution. In the Darwinian view, evolutionary progress is primarily based on variation of traits among individuals and natural selection of beneficial traits.

When studying biological evolution at the molecular level, a greater part of single nucleotide mutations are selectively neutral, i.e., they have no effect on the phenotypic expression. At a first level, the genetic code, and the translation of triplets of nucleotides to amino acids with certain characteristics is relatively error-proof – see for example the related work in (Bošnački, ten Eikelder, & Hilbers 2003). Silent mutations are mutations that do not change the resulting amino acid, and are possible since multiple codons exist for most amino acids. On a higher level, changing an amino acid in the peptide chain that makes up a protein, is in a large part of the cases not deleterious for the function of the protein. This second type of mutations, where the new amino acid is chemically similar to the one it is replacing, are called neutral mutations. Ng & Henikoff (2002) find that 75% of single nucleotide mutations do not lead to changes in the functionality of a protein. As a result of these available fail safe mechanisms in molecular biology, many mutations do not affect an individual's functionality, and are thus selectively neutral, or nearly-neutral, such that these mutations are not subject to (or insignificantly subject to) pressure of natural selection. Through genetic drift, new alleles (as a resulting product of silent or neutral mutations) may increase their frequencies in the population. Most probably, these mutations fade away and are lost, but they may also become fixated in the population, as a result of drift. As such, silent and neutral mutations can accumulate in the population, without affecting the fitness of the individuals, causing genomes and populations to evolve, complimentary to the Darwinian evolution of traits subject to natural selection. Note that we say that a population evolves if its allele frequencies change over time. The process of mutations that become fixated is thus also considered as evolution, even if the adopted changes are only apparent on the genotypic level.

The theory of neutral evolution can be adopted, for example, as a molecular clock, to determine how long isolated populations of the same species have been separated, as the rate of accumulated neutral and silent mutations is the same for both separated populations. Also, neutral and silent mutations may build up in a population as a hidden source of variation, letting the populations explore phenotypes which are otherwise unreachable, giving rise to neutral networks in phenotype space on which the population can evolve neutrally, see e.g., Eigen et al (1989) or Barnett (1997).

2.2 Research Fields of Interest

Models of infinite and finite population size have been studied in population genetics, evolutionary game theory and the theoretical study of the dynamics of evolutionary algorithms. In all three of these research fields, the development of models with finite population size emerges years, or even decades after the main principles have been laid down in models with infinitely large populations. We give an overview of these disciplines, and illustrate a selection of the main concepts and principles with infinite and finite population size, because of their relevance to this thesis. Note that this literature overview is concise, despite its length. The list of literature referenced is far from complete. The concepts and principles in this overview are selected based on their relevance to the rest of the text.

2.2.1 Population genetics

Theoretical population genetics studies the dynamics of frequencies of alleles in biological populations under the influence of evolutionary forces, such as natural selection and variation. Overviews can be found in Hartl (1997, 2000) and Ewens (2004).

Infinite populations. We shortly review three fundamental concepts of population genetics – Mendelian inheritance, the Hardy-Weinberg principle and Price's equation – and then elaborate on the fact that these concepts explicitly assume infinitely large population sizes.

Quantitative studies of dynamics in populations started historically with the experiments on the reproductive properties of peas by Gregor Mendel (1865). Mendel's observation showed that – after cross-pollination and repeated self-pollination of pea plants with two characteristics of a certain trait (i.e., flower color or seed shape and color) – the frequencies of properties of the trait consistently have a ratio of 3 to 1. In his studies, Mendel discovered that inheritable units are discrete characteristics – i.e., inheritable traits don't blend when parents' traits are recombined, as suggested by Darwin (1859) – and that each individual possesses two of these characteristics, i.e., they are diploid.

Only shortly after the rediscovery of Mendel's paper by Hugo de Vries and others in 1901, the first and possibly most important concept in population genetics, the Hardy-Weinberg principle, was independently discovered by Hardy (1908) and Weinberg (1908). A diploid population, whose ratio of AA:Aa:aa individuals is according to $p^2 : 2pq : q^2$, remains in this equilibrium if no evolutionary forces, such as selective or mutational pressure, act on the population.

Fischer's fundamental theorem of natural selection (Fischer 1930), and its generalization, Price's covariance and selection equation (Price 1970; Price 1972), state that the rate of increase of mean fitness over time is exactly equal to the variance of fitness in the population at that time. More generally, Price's equation expresses the rate of change of various characteristics of a population, of which mean fitness is just one.

All three of these important principles, which can be found in any population genetics text book are expressed or derived in a context of infinitely large populations, where populations are represented as collections of frequencies of individuals. Stochastic effects of random sampling on these principles are not directly implied, as the infinite population models are treated as deterministic models in their basic form. The concepts express the expected behavior of the systems, or typical expected populations, while ignoring the variances caused by random sampling in finite populations, and possible accumulation of these variances over time.

Finite populations. Early on, it was observed that fluctuations of allele frequencies appear in populations with finite population size, and that populations may become fixated in a certain allele. Finding the probability of fixation was first studied by Fischer (1922) and Haldane (1927), and later adapted on by Fischer (1930), who suggested the usefulness of models that assume finite populations to study this and similar problems.

Wright (1931) was, based on the ideas proposed by Fischer (1930), first to develop a finite Markov chain to model the dynamics of a finite population, containing individuals with one locus and 2 alleles for that gene. The models use the same assumptions as the Hardy-Weinberg principle – no selection or variation are present – with the exception of a finite and constant population size r. Each of the possible populations represents a state of the model. In a generation, parents are randomly selected from the population, and a new non-overlapping population of child individuals is produced. The probabilities that populations change their allele frequencies from one generation to the next, is written as a transition probability matrix T. Element T_{ji} represents the transition probability to move from a population with i instances of a certain allele to j instances of the allele in one generation. The transition matrix for one locus with two alleles is a matrix with $2r + 1$ rows and columns. The individuals are represented at their haploid state, by their gametes, therefore requiring twice as many (plus one) states as the size of the population of diploid individuals. If the distribution over the states of the system is given by $\mathbf{x}$, then the distribution at the next generation is given by $T\mathbf{x}$. Subsequent distributions can be used to study the dynamics of the population. See also Ewens (2004) for an updated mathematical overview of Fischer and Wright's results. We give an elaborate overview of these and similar models, and revise one of Wright's initial results in Chapter 4.

The Wright-Fischer model can be approximated by a diffusion model as developed by Kimura (1955, 1964). Kimura (1957, 1962) and Crow & Kimura (1970) have elaborately used these diffusion models to study the problem of fixation of alleles. Roughly summarizing their results, the fixation probability for deleterious alleles grows as the population size becomes smaller, which is, again, in line with the results of populations on which no selective pressure is exerted. Kimura's development of the neutral theory of molecular evolution follows from these results as discussed in Section 2.1.2.

2.2.2 Evolutionary game theory

Evolutionary Game Theory (EGT, overviews can be found in (Weibull 1995; Hofbauer & Sigmund 1998)) studies the dynamics and equilibriums of games played by populations of players. The strategies players employ in the games determine their interdependent payoff or fitness. In contrast with the traditional applications of game theory, the players do not act rationally when choosing their strategies (Dixit & Nalebuff 1993; Bierman & Fernandez 1997; Fudenberg & Tirole 1991; Fudenberg & Levine 1998), but act instead according to a preprogrammed behavior pattern. A strategy for playing the game is encoded in an individual's genome, which can evolve over time while repeatedly playing a game against other players in a population. Evolutionary game theory was introduced by John Maynard Smith & George Price (1973) and given full account by Maynard Smith (1982).

A game is represented by a payoff matrix A, where element A_{ij} represents the payoff received by a player employing strategy i when facing an individual adopting strategy j. The fitness of an individual is determined by the expected payoff received when playing the game against other individuals in the population. The reproductive success of an individual is thus not only determined by the individual's genome, but also by the frequencies of strategies in the population. We say that the fitness of an individual is frequency-dependent on the composition of the population. Note that when the population evolves, these frequencies change over time. Consequently, the expectations of payoff, and thus the fitness and selective pressure imposed on a strategy changes over time as evolution proceeds.

Infinite populations. A common model to study the dynamics of frequencies of strategies adopted by these populations is based upon replicator equations. A replicator equation is a system of differential equations, defined by the per capita growth of a strategy, according to the payoff received by the individuals playing the strategy. Replicator dynamics assumes continuous time, i.e., overlapping, infinite populations, asexual reproduction, complete mixing, i.e., all players are equally likely to interact in the game, and strategies breed true, i.e., strategies are transmitted to offspring proportionally to the payoff achieved. For an overview, see for example Hofbauer & Sigmund (Hofbauer & Sigmund 1998). Evolutionary game theory is closely related to agent-based computational economics, see e.g., Tesfatsion (2001) as it can serve as a model for the dynamics in social and economic simulations.

A central concept of EGT is the evolutionarily stable strategy (ESS), first proposed by Taylor & Jonker (1978). If a population of individuals all play a certain strategy, and it can not be invaded by any mutant strategy, because this mutant strategy would be selected against in competition with all other individuals, then the strategy used by all of the individuals in the population is an ESS. The ESS is closely related to the Nash equilibrium in classic game theory. If players are assigned a strategy for a game such that no player can benefit by changing his strategy while the other players keep their strategies unchanged, then that set of strategies constitute a Nash equilibrium.

Finite populations. The ESS is described in a context of infinitely large populations. Indeed, as small populations are considered, there is a small, but non zero probability that the mutant strategy takes over the population that plays the ESS. Indeed, there is a small non zero probability that a population of deleterious mutants is sampled within a generation, instead of a population of ESS players, because of stochastic sampling. This probability is small, but increases as the size of the population decreases. Consequently, the important concept of evolutionary stability of strategies does not take stochastic sampling effects in small populations into account.

Foster & Young (1990) were first to question the ESS criterion in stochastic environments. The ESS principle says that a small divergence of the ESS will eventually die out. However, a system doesn't always return to the initial state if a perturbation, e.g., a mutation to a deleterious allele or strategy – occurs. Even in an infinite population with variation, the ESS remains under pressure of continuous perturbations. Instead of the ESS, Foster and Young propose stochastically stable equilibriums. A state is a stochastically stable equilibrium if, in the long run, it is almost certain that the system is in the neighborhood of the state, as stochastic noise – i.e., variation through mutation or stochastic effects of finite populations – goes to 0. As an example, Foster and Young consider the game with payoff matrix

$$A = \begin{pmatrix} 4 & 0 \\ 0 & 8 \end{pmatrix}.$$

This type of game, which only has strictly positive payoffs on its diagonal, is called a coordination game, with both of the strategies being Nash equilibriums. A single mutant individual in an otherwise uniform population has a very small chance of survival. Both strategies are an ESS of the coordination game. However, if variation is considered, then populations involved in this game are mainly attracted to adopt strategy 2. Foster and Young show that populations with the second strategy are the stochastically stable equilibrium of the coordination game. The authors make the observation that including stochastic perturbations has its consequences for understanding the stability of evolutionarily stable strategies. However, the proposed method (via minima of potential functions) is not practical when concrete strategies are studied and the influence of their stochastic perturbations must be interpreted. The authors conclude that it would have been interesting to study the stochastically stable equilibriums for specific data of mutation rates and variations of reproductive success. In Chapter 6, we study such particular games and their strategies, although finite Markov models are adopted for the study instead of the proposed use of potential functions.

Games with a balance between payoff dominance and risk dominance have been used to study the effects of stochasticity on the evolutionary stability of strategies. As an example, consider the game with payoff matrix

$$A = \begin{pmatrix} 5 & 1 \\ 3 & 4 \end{pmatrix}.$$

Both strategies are, in similarity with the coordination game, an ESS of the game. In this game, strategy 1 payoff dominates strategy 2, since with strategy 1, more payoff can be earned than with strategy 2. On the other hand, strategy 2 risk dominates the first strategy, as the risk of losing payoff as a result of mutations in the population is higher for strategy 1 (payoff could drop from 5 to 1) as compared to the risk involved when opting for strategy 2 (payoff difference is 4 to 3). Games with risk dominance, like the one given above, is the subject of Harsanyi & Selten (1988). Kandori, Mailath & Rob (1993) and Robson (1996) have studied effects of finite population size on the ESS of games with risk dominance. Mię̧kisz (2005a) has studied the limit behavior of finite population models in these games, for a set of cases with specific generational transition rules. The main conclusion is the observation that for an arbitrary low mutation rate, and a small population size, the population prefers the strategy with risk dominance. As the population size increases, the population undergoes a transition from choosing for the risk dominating strategy to the payoff dominating strategy. Mię̧kisz concludes that for studying a specific model and the limit behavior of a system, it is important to take the population size and mutation rate into account. We investigate this conclusion in Chapter 6. Mię̧kisz has also extended this study to other games, such as spatial games and games with more than 2 players, see e.g., Mię̧kisz (2005b, 2004a, 2004b, 2004c).

Fogel et al (Fogel & Fogel 1995; Fogel, Fogel, & Andrews 1997; Fogel, Andrews, & Fogel 1998) and Ficici & Pollack (2000a) have studied finite population effects of evolutionary dynamics on the stability of evolutionary stable strategies of the Hawk-Dove game (see Chapter 6) empirically. Using simulations of the evolutionary systems, behaviors have been observed that are unrelated to an evolutionary stable strategy. They have suggested that ESSs may not provide a good expectation of a finite population's behavior. In Chapter 6 we present an exact Markov model approach to answer the same questions for a larger set of games, adapting Ficici's initial work (Ficici & Pollack 2000a). We adopt genetic drift, and the causes of genetic drift, as an explanation of these observations.

Nowak & Sigmund (2004) have also expressed the importance of finite populations when evolving strategies for games. Nowak et al (2004) have used a Moran process (Moran 1962) to model the behavior of a population involved in the iterated prisoners' dilemma, which also incorporates the time to fixation and invasion coefficient of strategies in finite populations. See also (Taylor, Fudenberg, Sasaki, & Nowak 2004) for more applications of this model for different games. Nowak et al have shown that within a finite population, cooperative strategies – a strategy that is at cost of the individual, but beneficial for its contenders – may invade a population of non-cooperating individuals, given that the population is relatively small. This is in contrast with the common belief of evolutionarily stable strategies, which says that defectors are stable against invasion by cooperators. Cooperation, however, is fairly common in nature, and finite population sizes may explain how a cooperating strategy can become fixated through random sampling in a small population. In similarity with their models, Nowak et al suggest the use of Wright-Fischer models to further study the behavior of finite populations in cooperation-defection games. We provide such a study of the prisoners' dilemma in Chapter 6.

It should be noted that the study of finite population effects in evolutionary game theory has only been established in the recent years.

2.2.3 Evolutionary algorithms

In evolutionary algorithms – such as Genetic Algorithms (GAs), evolution strategies, or genetic programming – heuristics to find approximate solutions of optimization problems are inspired by evolutionary mechanisms such as selection, inheritance and variation through recombination and mutation (Goldberg 1989; Mitchell 1996; Koza 1992). The typical implementation is a computer simulation of an evolving population of candidate solutions for a given optimization problem. Candidate solutions are selected from the population according to their fitness, and selected individuals are coupled to produce a new generation of candidate solutions through recombination (crossing over) and variation (mutation). This generational process is then repeated until a satisfactory solution for the problem is produced, or another termination condition is fulfilled.

The objectives of the theoretical research of evolutionary algorithms are commonly different from the goals in population genetics. Studies in population genetics mainly focus on studying the dynamics of a given population with a predetermined reproduction scheme, i.e., including selective pressures and mutation rates. Since GAs are mainly applied in engineering environments, theoreticians in the GA community search for optimal parameters of the optimization process – rates of mutation and selection schemes – such that the optimization algorithm can find (near-)optimal solutions of combinatorial optimization problems in the least amount of time. In the study of GAs, time is generally expressed in numbers of fitness function evaluations. Indeed, in practical optimization problems, the evaluation of fitness – required for selection – is usually computationally expensive. This is in contrast with population genetics, which measures time in numbers of generations. In nature, the determination of biological fitness is in parallel, where the evaluation of fitness in GAs is a computationally costly task. As a result, parameters in GA models may differ greatly from the parameters in population genetics. As an example, the rate of mutation in population genetics is usually low enough (close to 0) such that it may be ignored, or that the limit as the mutation rate goes toward 0 is usually sufficient to make predictions. In GAs, mutation rates are generally higher. Similarly, a GA practitioner has to decide on population sizes and selection schemes to devise algorithms with the least amount of fitness evaluations. As another contrast with research in population genetics, GAs have been developed to solve optimization problems that have multiple loci and are hard because of deceptivity and high rates of interactions among the loci, see e.g., Naudts and Verschoren (Naudts & Verschoren 1999), Naudts (1998) and Suys (1999). As a consequence of these differences, models used in the study of population genetics is not always applicable for adoption by GA practitioners, and specific adaptations of these models are required to serve these specific goals.

Infinite populations. In the study of the dynamics of genetic algorithms, Holland (1975) has proposed the schema theorem as a foundation to express the power of genetic algorithms with bit string representations of candidate solutions. The schema theorem says that schemata (patterns of bit strings) with above-average fitness increase in frequency in the genetic algorithm over time, thereby explaining the power of a genetic algorithm. Many criticisms have questioned the explanatory capabilities of the schema theorem. Radcliffe (1992), for example, discusses instances of specific problems with non-linearity between the genotype space and phenotype space, showing that schemata were unable to describe the dynamics in the genetic algorithm. Altenberg (1995) gives an overview of these critiques and shows how Price's equation can be used to adapt the schema theorem, and to express the performance of a GA in terms of mean fitness, instead of in the growing frequency of schemata with above-average fitness.

Vose & Liepins (1991) have constructed a dynamical systems model to study the dynamics of a GA, using the infinitely large populations to approximate the dynamics of finitely large populations. A population is represented by its constituent frequencies of individual types from the search problems' search space, thus represented by a stochastic vector. Selection and recombination operators are mathematically represented with matrices, and the eigenvalues and eigenvectors of the matrices can be used to study the fixed points of the populations' search space. The selection and reproduction operators acting upon the infinite populations are deterministic, giving a unique population in each of the generations for an initial population that starts the chain. In the limit of increasingly larger populations, this model is thus mathematically exact. For a more elaborate account, see Vose (1999b).

Finite populations. In the genetic algorithms community, there is little awareness of the effects of random genetic drift in the evolution with small populations. Populations of 1000 individuals are generally considered large in this community, and the average population size used by practitioners is around 100. This number is an estimated median of the population sizes used in papers of the real world applications track (RWA) in the proceedings of GECCO 2004 (Deb 2004). Around 50% of the applications in this track use a population size of 100 or smaller.

This number is very small in relation to the search spaces of the examined optimization problems, and not considered as being large for a population geneticist. In evolutionary algorithms with small population sizes, the algorithm can converge prematurely. Because of lack of diversity, caused by exploiting a small sample of relatively fit individuals in the initial generations of the search process, the population gets fixated, or stuck in local optima of the search space. As populations become smaller, these effects of random genetic drift in evolutionary search in an optimization problem is indeed more evident, affecting the algorithm's performance.

Nix and Vose (Nix & Vose 1992) developed, as a follow-up to Vose and Liepins' infinite population model, a finite population version of the dynamical systems model. The finite Markov model is described in large extent in Vose (1999b), Rudolf (1998) and Schmitt (2001). The model is similar to the Wright-Fischer model (Wright 1931; Fischer 1930), but has been developed independently. The Nix and Vose model assumes – in contrast with the Wright-Fischer model – haploid, multi locus individuals.

Similarly to the Wright-Fischer model, each possible population configuration of a given population size is represented by a state of the finite Markov model. The transition probability matrix describes the probabilities to go from one state – a population with a given number of instances of each genotype – to any other. The transient and limit behavior, described implicitly by the transition probability matrix, of a distribution over all states can be studied to provide insights on the influence on the long run behavior of population size, and other parameters of the evolutionary model.

Note that since all possible population compositions must be accounted for in this model, the resulting transition probability matrices easily become too large to handle. The study of the models is consequently limited to small genotype search spaces and small population sizes. As a consequent concern, it is hard to discuss the scalability of results obtained from these models, for increasing problem and population sizes. We discuss this problem, and review possible solutions for compressing the transition probability matrices in Chapter 3.

De Jong et al (1995) provide an initial analysis of transient behavior in these models for specific (small) optimization problems, in order to understand the properties of genetic algorithms being used as function optimizers. They conclude that an exact closed form analysis of the Markov models is generally too hard, because of the complexity in constructing these models. However, the experimental mathematics approach taken provides insights into the workings of a GA. The methodology of this thesis follows a similar approach to studying the limit behavior of evolutionary systems.

Van Kemenade et al (1998) construct transmission function models to study the importance of several stochastic elements in a genetic algorithm. They confirm the correctness of their model by analyzing varying selection methods and conclude that the finiteness of a population strongly influence the behavior of a genetic algorithm.

Vitányi (2000) adopts finite Markov models to suggest a method for constructing evolutionary systems with small population size that have a high probability of converging to the optimal solution of an optimization problem in polynomial time. The construction requires rapid mixing of the Markov model representing the evolutionary search algorithm, i.e., the second largest eigenvalue of the probability transition matrix should be suitably bounded away from 1, and the Markov process must have a steady state distribution of the process with a high probability of populations with optimal solutions. In contrast with De Jong's work, the author provides a formal analysis of an example evolutionary system, instead of relying on the experimental mathematics approach. It is however difficult to construct an algorithm that satisfies both conditions, but the proposed methodology formally paraphrases the requirements for a good design of an evolutionary algorithm – high speed of convergence to a high probability of optimal individuals – in terms of finite Markov models of evolutionary algorithms.

Vose & Liepins (1991) note that fixed points of the model outside the space of the model govern the transient dynamics of the evolutionary algorithm. The model is attracted to these fixed points before continuing its path to the fixed points within the state space. This gives rise to periods of stasis in the evolutionary dynamics, intermitted by rapid innovations, i.e., these external attracting states lay at the basis of the punctuated equilibria in evolution. Van Nimwegen et al (1997) show that this effect is further amplified in finite populations, and identify the epochs of stasis with the flow's metastable fixed points, giving exact predictions of stasis. Van Nimwegen et al (1998, 1999) also provide further analysis, and analyze the punctuated equilibria behavior of a genetic algorithm with the royal road fitness function. It is concluded that epochal evolution is the result of interplay between the finite population flow given by the heuristic function of the infinite population model, and coarse-graining of the state space because of finite population size. The authors also relate these results to the predictions of the neutral theory of molecular evolution of Kimura (1986).

Recognizing the complexity of enumerating all populations in a Markov model, Prugel-Bennett (2003) has proposed statistical mechanics approaches to studying finite population models, by giving concise descriptions of the distributions. Although the descriptions are approximations of the exact behavior of the systems, the models provide an accurate description of the behavior. It is concluded that infinite population models can provide worse descriptions of the behavior than the proposed models, even for very large populations, for example for deceptive and epistatic fitness functions that make optimization problems hard for genetic algorithms.

Alkemade et al (2005) provide a first step in relating the processes in agent based economics with evolutionary algorithms. In economic modeling, evolutionary algorithms are often used to simulate the evolution of economic strategies, although the realism of relating a social or economic simulation to an evolutionary algorithm is often questioned (Chattoe 1998). The parameters of the evolutionary algorithm – such as population size and learning rates – are often directly related to the economic model studied, but they note that small population size of the economic model hinders the learning in the population. The authors conclude that parameters from agent based models should be treated separately to construct robust simulations of agent based models with evolutionary algorithms. This observation is similar to the observations made of random sampling being the cause of genetic drift in finite evolutionary models. It would be interesting to investigate how effects of random sampling in economic and social studies would relate to the effects of random genetic drift in evolutionary systems, adopting Markov models similarly to the methodology of finite evolutionary systems, by extending the bounded rationality present in Darwinian selection.

2.3 Outline and Methodology

In this thesis, we develop finite Markov models of finite haploid and diploid populations in dynamic, game theoretic and co-evolutionary fitness environments. Using an experimental mathematics approach to study the models, we analyze the long run behavior of the systems by computing the limit distributions of the Markov models for specific parameters, such as population size, dynamic characteristics of selective pressure, variational pressure, ploidy and dominance. We compare the predicted behavior of the finite population models with predictions of corresponding infinite population models, which allow us to discuss the influence of finite population size on the behavior of populations in dynamic fitness environments. The following sections elaborate on the concepts that constitute our methodology, and provide an outline of the text.

2.3.1 Finite Markov models of finite populations

A stochastic process representing an evolutionary system is a sequence of random events or variables in time

$$P(0) \to P(1) \to P(2) \to \dots \tag{2.1}$$

where each of the random variables $P(g)$ represents a population, i.e., a state of the system, at time step or generation g. The population, or state in which the system resides at one time step is influenced by the previous states of the system. Moreover, in an evolutionary system, the population $P(g)$ at one time step only depends on the population $P(g-1)$ of the previous time step. We say that there is no memory of earlier time steps in the sequence. Such a system, without memory, is a Markov process, and the sequence of states is a Markov chain.

In a deterministic model with infinite populations, $P(g)$ is uniquely defined by its predecessor $P(g-1)$. In a stochastic system, there is a probability distribution over all states, which is a function of the state in the previous time step. With finite populations, defined over a finite genotype space, the state space π of all populations is finite, and the Markov chain is consequently said to be finite. For each population $P \in \pi$, a distribution over π describes the probabilities to transit in one time step from P to any of the populations in the next time step. We can describe the system with a $|\pi| \times |\pi|$ transition probability matrix T, where a row or column corresponds to a state P. A column gives the distribution over the states at the next time step if the current population is P.

If $\mathbf{x}(g)$ is a probability distribution over the states – i.e., a stochastic vector of size $|\pi|$ – then the distribution at the next time step is given by the discrete version of the Chapman-Kolmogorov equation (Papoulis & Unnikrishna 2002)

$$\mathbf{x}(g) = T\mathbf{x}(g-1) = T^g\mathbf{x}(0). \tag{2.2}$$

The distribution over the states at time step g is thus uniquely defined by the transition probability matrix T and the initial distribution $\mathbf{x}(0)$ over the states at time step 0.

If the transition probability matrix T of the Markov model is ergodic, than there exists a unique stochastic eigenvector $\mathbf{x}^*$ of matrix T, with corresponding eigenvalue 1, according to the Perron-Frobenius theorem, see Appendix B. This stochastic eigenvector describes the limit, or steady state distribution over the states of the system, independent of the initial distribution over the states of the system. The limit distribution thus gives the probability for each of the states of the model, that the system ends up in this state. The limit behavior of a finite Markov model can be adopted to analyze the distribution over the states of the system in the long run. By properly analyzing the eigenvector, with corresponding eigenvalue 1, of the transition probability matrix, information can be gained on the behavior of evolutionary systems.

2.3.2 Ploidy

The models we develop in this text have their applications in both population genetics, game theory and genetic algorithms. Generally, population genetics is based on the reproduction scheme of higher order species. Consequently, their models represent the evolution of populations of diploid individuals. This is in

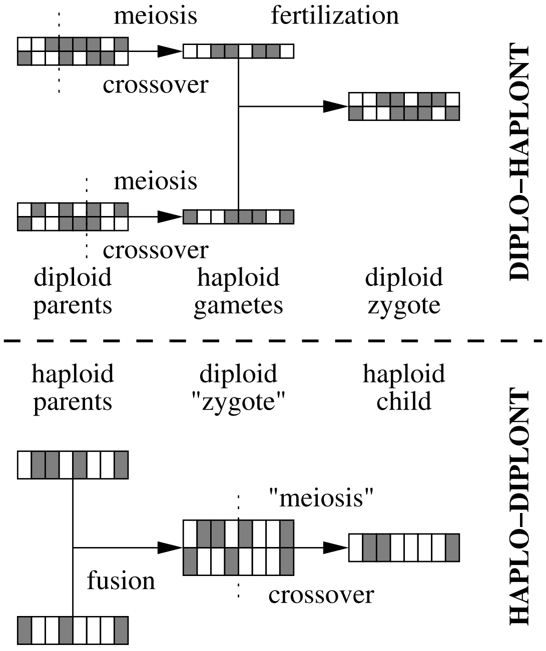


Figure 2.2: Schematic representation of sexual reproduction in diplo-haplont (top) and haplo-diplont life cycles.

contrast with evolutionary game theory and genetic algorithms, where individuals are generally assumed to be haploid. Since both haploid and diploid populations are of relevance to this text, we construct finite Markov models of both haploid and diploid populations in Chapter 3. Here, we shortly review haploid and diploid reproduction for the reader who is unfamiliar with ploidy.

In population genetics, each diploid individual has two homologous chromosomes, or twice the amount of genotypic information when compared to haploid individuals. Such single stranded individuals are the subject of research in genetic algorithms. For a certain gene, a diploid individual thus stores two, possibly different alleles. The alleles may thus disagree on the outcome of the phenotypic expression of the gene. Such conflicts are resolved in biological individuals on a biochemical level. Because of cascades of chemical reactions and reactants, alleles in both genomes are expressed, but their quantitative influence on the phenotypic expression differs. This results in a dominance and recessiveness relation between homologous alleles. If both alleles are similar, or homozygous, the phenotypic result is indisputable, but when they differ, i.e., when the alleles are heterozygous, their phenotypic expression is determined by their dominance relation. As an example, consider the gene determining eye color. The allele for brown eyes dominates the allele for blue eyes. Consequently, the allele for blue eyes can only be expressed if it is not paired with an allele for brown eye color.

When two diploid individuals are selected to interact in sexual reproduction, both individuals create haploid gametes through the process of meiosis, a cellular process that forms the basis of sexual reproduction. During meiosis, the individuals undergo gene duplication – which might cause mutations in their genotypes – and chromosomes are split up and merged together through the process of crossover. Next, the cell divides into 4 haploid daughter cells, each containing half a (mutated) copy of the original parent's genes, completing meiosis. When two haploid gametes are joined together through fertilization, a new diploid child of the original parent individuals is formed. This child is a member of the population of the next generation in the evolutionary system at hand. This life cycle of individuals that are diploid at maturity, i.e., at the time of fitness evaluation, is called the diplo-haplont life cycle.

In the life cycle of a haploid individual, two individuals are fused, after which meiosis takes place to create new haploid individuals. This reproduction scheme of the haplo-diplont life cycle forms the general principle of reproduction in genetic algorithms and evolutionary game theory.

Figure 2.2 depicts the haplo-diplont and diplo-haplont life cycles schematically. Note that both life cycles contain the same elements, i.e., fusion or fertilization of haploid gametes and meiosis of the diploid zygote to produce daughter haploid gametes. The difference between the two models is the state in which the individuals are mature, i.e., when fitness is determined.

Diploidy in genetic algorithms. There have been several occurrences of adopting diploidy in implementations of genetic algorithms, breaking with the haploid dogma of genetic algorithms. Goldberg & Smith (1987, 1992) suggested dominance and diploidy as a means for dynamical optimization, and reported the first experimental and theoretical results. Banzhaf (1988) also reports experimental results and discusses the maintenance of variability in diploid genetic algorithms. Yoshida & Adachi (1994) and Collingwood et al (1996) discuss empirical results along the same line. Hillis (1992) utilizes a specific "diploid" genome to evolve sorting networks in a co-evolutionary environment. Hadad & Eick (1997) study polyploidy and dominance vectors and discusses empirical results for dynamical problems. Lewis et al (1998) apply diploidy to non-stationary problems and discuss their performance. Ryan & Collins (1998) discuss the performance of haploid and diploid search algorithms.

A couple of formal models were also suggested to study some topics relating to diploidy in genetic algorithms. Bidwell (1996) and Wright & Bidwell (1996) have built and discussed models similar to the ones used in population genetics. Greene (Greene 1999) studies diploidy in deceptive environments using a formal model which offers some results, but at the cost of using a model with populations of infinite size.

As can be seen from this literature overview, there have been two main methods of studying diploidy and dominance in genetic algorithms; (1) either to study empirical results obtained from diploid implementations and compare these with haploid genetic algorithms or (2) to implement a formal, infinite size model and discuss its behavior. In this thesis, we construct a finite population size model wherein diploidy and dominance can be studied, and with which exact results can be obtained, without having to rely on empirical experiments.

2.3.3 Dynamic environments

Note that in the basic form of the Wright and Fischer model, or the Nix and Vose model, the fitness determination of individuals is static, i.e., the selective pressure of alleles is constant over time. In practice, genetic algorithms indeed commonly address optimization problems that do not change over time. Population geneticists traditionally study fitness determining environments in which selection is fixed over time. Many real world applications of optimization techniques, and natural biological environments, however, have characteristics that change over time while the evolutionary system is trying to optimize for these changing environments.

Throughout this thesis, we construct models of finite populations in dynamically changing fitness environments by combining finite Markov models of finite populations in different static fitness environments. Environmental dynamics can be established in evolutionary models by either explicitly or implicitly forcing a fitness function's dependence on time. In the explicit case, we let the fitness function be dependent on the number of generations, and in the implicit case we let the selection be dependent on the composition of the population. Since this population changes over time as a result of the evolutionary forces acting on the population, it indirectly generates time dependence of the fitness values.

In the explicit case, we let the fitness function change over time by altering the fitness of individuals over time, independent of the dynamics of the evolutionary system. In Chapter 5 we develop finite Markov models with dynamic fitness environments by imposing alternating fitness environments on the selective process of the evolutionary system. We let a static fitness function govern the evolutionary system for a number of generations, after which another fitness function takes over the selection process of the evolutionary system. By alternating such a set of fitness functions over time – either deterministically, or stochastically by modeling the fitness environment as a Markov chain on its own – we can model an evolutionary system that assumes a dynamic fitness function, and model this whole system as a Markov process. We show the conditions for which such systems are ergodic, and study, in the case of ergodicity, the long run behavior of the system, by examining the Markov chain's unique limit behavior.

By defining the fitness evaluation as a function that is dependent on the frequencies of individual types in the population, we can implicitly impose dynamics on the fitness function. In evolutionary game theory, the fitness, or expected payoff, of an individual is dependent on the composition of the population. Similarly, fitness functions with fitness sharing are dependent on the composition of populations, and have been studied extensively by Horn (1997). As the population changes over time because of evolutionary forces acting upon the population, the expected payoff received by a strategy also changes over time. We study models of finite populations with fitness functions according to evolutionary games in Chapter 6.

Another way to conceive a dynamical fitness environment for a population, is to let it co-evolve with another population. In co-evolution, two or more populations evolve simultaneously, and the fitness function of individuals in one population is dependent on the composition of the other population. Co-evolution is thus defined as reciprocal evolutionary change in interacting species. Similarly to the argument used for frequency dependent fitness, evolutionary change in one population directly brings changes about in the selection process of the other population. Consequently, populations are put against each other and govern each other's fitness function dynamics. Models of co-evolution that assume finite population size are developed and studied in Chapter 7.

We have to point out that the term "co-evolution" is disputed over in the genetic algorithms community. Fitness dependence of an individual on other individuals in its own population is also considered as co-evolution by some authors, see e.g., Ficici (2004). In this text, we make the distinction and say that a co-evolutionary system basically requires more than one population, following the biological definition, and we do not classify frequency dependent fitness functions under co-evolution.

Ploidy in dynamic environments. Diploid individuals store redundant information in the recessive part of their genome. These recessive alleles are not subject to selective pressure, and the collection of recessive alleles in a population can consequently serve as a pool of deleterious alleles, remembered by the diploid population. If a diploid individual is created that contains two such recessive alleles, this allele is expressed and becomes subject to selection. Diploidy and the process of dominance thus offer a mechanism to store alleles in the recessive pool of a population, and these recessive alleles can be retrieved from this pool to test their fitness.

In dynamic fitness environments, this recessive allele pool can be thought of as an implicit memory of solutions for previously encountered problems. If the environment returns to such a previous state, then a diploid population may retrieve the solution from its allele pool, if present, by combining recessive alleles. In contrast, these solutions of previously encountered environments are forgotten in a haploid population as the alleles in a haploid population are constantly subject to selective pressure, removing deleterious alleles, and thus solutions from the past, from the population. Consequently, diploid populations are hypothesized to perform better than haploid populations in dynamic environments.

Additionally, the storage of redundant information slows down convergence of a population during optimization, but at the same time it can keep the diversity of individuals – and thus the variance of fitness – high. According to Fischer's fundamental theorem of population genetics, and Price's equation of covariance and selection, the rate of increase in mean fitness of a population is higher as the variance in fitness in the population is higher. Consequently, a diploid population that maintains a high diversity can be beneficial when constructing an optimization algorithm.

The use of diploid genetic algorithms in non stationary fitness environments was first suggested by Goldberg et al (1989) and further investigated by Ng & Wong (1995). As an example dynamic fitness function, the authors consider a dynamic version of the $0-1$ knapsack problem. In a knapsack problem, items of different size need to be selected from a big list of items, and stored in a limited amount of space, i.e., a knapsack. The goal of the problem is to find those items such that the knapsack is as much filled as possible. In the dynamic version of the knapsack problem, the size of the knapsack is changed over time, as the algorithm proceeds, and items need to be constantly removed, or added to find the collection of items that best fits the knapsack at that time. Empirical observations with haploid and diploid versions of the genetic algorithm have indicated that diploid populations are useful in periodic environments, i.e., where the size knapsack is alternating. In these periodic environments, only a few states need to be remembered by the diploid allele pool and solutions to previously encountered problems can be retrieved quickly.

For some dynamic problems in this thesis, we test the hypothesis that diploid populations can perform better in small dynamical problems, by comparing the performance of haploid and diploid populations in dynamic fitness environments. In Chapter 7, we let a haploid and diploid population co-evolve, using co-evolution as a test bed to confront haploid and diploid populations in a dynamical environment.

Chapter 3

Models of Haploid and Diploid Populations

In this chapter, we develop discrete time Markov models of haploid and diploid simple genetic algorithms, with finitely and infinitely large populations. We introduce the simple genetic algorithm in its haploid and diploid context. We review concepts of a search space and population space. We discuss the processes of selection and reproduction. All of these elements are then combined to define the transition probabilities between states, or populations, of the SGA. Numerical methods for finding the limit behavior of the resulting Markov models are discussed. Methods for computing the steady state and limit behavior of the systems are discussed. The resulting models act as building blocks for constructing more complex models to serve the applications in later chapters.

The notation introduced in this chapter is used throughout the whole text and is based on the notation of Vose (1999b) but has been adapted to deal with the specific dynamic environments in the later chapters of this thesis.

Parts of this chapter are derived from A.M.L. Liekens, H.M.M. ten Eikelder, P.A.J. Hilbers, *Modeling and Simulating Diploid Simple Genetic Algorithms* (2003e).

3.1 The Simple Genetic Algorithm

The Simple Genetic Algorithm (SGA), or generational genetic algorithm is an abstraction of the genetic algorithm (GA). In its simplest form, the GA generates consecutive non-overlapping populations of individuals; the SGA assumes that a population is constructed from scratch at each generation. All of the individuals in the current generation's population are replaced by new individuals through processes of reproduction and selection, based on the fitness of the individuals in this population. The quality of individuals in these generations *evolves* over time based on the Darwinian principles of variation, selection and heredity of traits. The discrete time assumption of non-overlapping generations lays at the basis of our models of evolving populations.

Vose (1999a, 1999b) classifies the SGA as a special case of random heuristic search. In random heuristic search, the optimization process in a search space Ω is a stochastic chain of consecutive collections of elements in Ω.

In the SGA, an initial population $P(0)$ of random candidate solutions for the optimization problem starts the algorithm. With *candidate* solutions for an optimization problem, we denote solutions that are not necessarily optimal. A population $P(g + 1)$ at time step, or generation $g + 1$ is then stochastically produced from the previous population $P(g)$. Individuals are selected from the population according to their fitness. Abstract genetic operators, such as recombination and mutation are performed on these *parent individuals* to beget new *child individuals*. These children are placed in a new population that makes up the new generation. This generational process is repeated until some termination condition is fulfilled. Examples of termination conditions for genetic algorithms are a limiting number of generations or computational time used, the detection of convergence of the population, or finding a sufficiently good solution for the optimization problem. The morphology of the SGA is outlined in pseudo code in Algorithm 3.1.

Algorithm 3.1 (Simple genetic algorithm). Pseudo code of a simple genetic algorithm.

```
$generation := 0$
create initial population $P(generation)$ over $\Omega$
repeat
  repeat
    select $mom, dad$ from $P(generation)$
    $child :=$ reproduce$(mom, dad)$
    add $child$ to $P(generation + 1)$
  until $P(generation + 1)$ is filled
  $generation := generation + 1$
until terminated
```

As the SGA generates a stochastic chain of populations at discrete time intervals, where each new population is only dependent on the state of the previous population, the resulting model is a finite, discrete time Markov chain. Markov chain theory can be applied to the model in order to study the SGA's behavior.

3.2 Search Spaces

Two specific types of search spaces are of particular interest in this thesis. On one hand, we evolve haploid string individuals, following the genetic algorithm's dogma of haplontic life cycles. Secondly, we focus on models with diplontic life cycles of diploid string individuals. Both interpretations require specific search spaces. For the purposes of this text, we assume string representations for the genomes as a sufficient means to study the problems in the following chapters. Other common genome representations used in the GA community are, e.g., graphs, or trees in genetic programming, see (Cramer 1985; Koza 1992; Langdon & Poli 2002).

Generally, we use notation Ω to denote the search space of the genetic algorithm. However, we make a distinction between the search space of the heuristic, which is denoted by Ω_S, the search space of haploid genomes Ω_H and the search space of diploid genomes, Ω_D. We introduce these different search spaces in this section, and discuss the mapping used to transform elements from Ω_H or Ω_D to elements of Ω_S. If notation Ω is used in the construction of the model of the genetic algorithm, then both Ω_H and Ω_D are meant as possible search spaces for the genetic algorithm. Only if the distinction between haploid and diploid search spaces is required, their specific notations are used.

We assume that search space Ω_S of the heuristic is finite, with size $|\Omega_S| = n$. For ease of notation, we let each element $s \in \Omega_S$ correspond with a number, also denoted by s, with $0 \leq s < n$.

3.2.1 Strings

Let $s = (s_0, s_1, ..., s_{l-1})$ denote a string of length l. Each of the characters s_i is chosen from an alphabet Σ_i. Search space Ω_S is the set of all possible strings of length l, i.e., the Cartesian product $\Omega_S = \Sigma_0 \times \Sigma_1 \times \ldots \Sigma_{l-1}$. In many cases, the alphabet for any of the characters is the same. As an example, we can assign the set $\mathbb{Z}_2 = \{0, 1\}$ to alphabet Σ_i for all $0 \leq i < l$. Each string s with characters in alphabet $\mathbb{Z}_2$ then represents a bit string with length l and search space Ω_S becomes the set of all bit strings given by the Cartesian product $\Omega_S = \mathbb{Z}_2^l$. As another example, alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$ can be considered to model RNA molecules of length l, with $\Omega_S = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}^l$.

3.2.2 Haploid genomes

For the search spaces of haploid genomes, we assume that there is no specific genotype-to-phenotype mapping from haploid individuals to strings in the search space of the heuristic. As a result, a string, or genome s in the haploid search space Ω_H corresponds to the same element s in search space Ω_S, giving $\Omega_H = \Omega_S$. The position i in a string is called a *locus* i in the genome, and alphabet Σ_i is the set of *allele*s for that locus.

3.2.3 Diploid genomes

In the diploid model, an individual consists of two strings. An individual of the diploid population is represented by a multiset of two elements of Ω_S, i.e., $\{s, t\}$ with $s, t \in \Omega_S$. Note that a multiset is required since $s = t$ is allowed. The set of all possible diploid genotypes is denoted by $\Omega_D = \{\{s, t\} | s, t \in \Omega_S\}$, the search space of the diploid genetic algorithm. Since the search space is a set of multisets of cardinality 2 over Ω_S, the size of this search space is equal to

$$|\Omega_D| = \left(\begin{array}{c} |\Omega_S| + 2 - 1 \\ 2 \end{array} \right) = \frac{|\Omega_S| \, (\, |\Omega_S| + 1 \,)}{2}.$$

The number of multisets of cardinality c over a set of size n can easily be computed. Consider the notation for multisets of the form $\bullet\bullet \mid \bullet\bullet \mid\mid \bullet$ that would represent multiset $\{A, A, B, B, D\}$, a multiset of cardinality $c = 5$ over set $\{A, B, C, D\}$ of size $n = 4$. The number of multisets is the number of ways to arrange c bullets (or $n-1$ vertical lines) among $c+n-1$ positions, which leads to the resulting binomial coefficient $\binom{c+n-1}{c} = \binom{c+n-1}{n-1}$ denoting the number of multisets, see e.g., Mathews (1964) pp. 376–377.

If we assume that Ω_S is the search space of bit strings with length l, i.e., $\Omega_S = \mathbb{Z}_2^l$, then the search space of diploid bit strings Ω_D contains all multisets of two bit strings of length l with

$$|\Omega_D| = \frac{2^l\left(2^l+1\right)}{2} = 2^{2l-1} + 2^{l-1}.$$

Mappings from diploid individuals in Ω_D to elements in Ω_S are discussed in the following sections.

3.2.4 Dominance mappings

In order to map diploid genotypes to (haploid) phenotypes we have to define a dominance mapping, or operator $\delta : \Omega_D \rightarrow \Omega_S$ to act on the diploid individuals and which results in an element of the search space. Such a dominance operator allows us to reuse cost or fitness functions from the optimization problem in the diploid genetic algorithm.

Properties of dominance mappings. In practice, an operator $\odot : \Omega_S \times \Omega_S \rightarrow \Omega_S$ is chosen as the dominance mapping operator, such that $\delta(\{s,t\})$ can be computed using this operator with $\delta(\{s,t\}) = s \odot t$. In this thesis, we assume that 2 properties hold for $\odot$. First of all, we want homozygosity of homologous alleles to produce a consistent phenotypic expression, i.e., the dominance relation should be neutral with

$$\forall s \in \Omega_S : \delta(\{s,s\}) = s \odot s = s. \tag{3.1}$$

More pressing however, is the commutativity of operator $\odot$, since the strings that are presented to the operator originate from a multiset, whose elements have no ordering, i.e.,

$$\forall s,t \in \Omega_S : \delta(\{s,t\}) = s \odot t = t \odot s. \tag{3.2}$$

Locus-wise dominance operators. In the case of diploid bit strings, where an individual is a multiset of 2 bit strings, the dominance mapping operator can be defined as a bitwise operator. Properties (3.1) and (3.2) hold for bitwise operators AND (with $\delta(\{s,t\}) = s \wedge t$) and OR (with $\delta(\{s,t\}) = s \vee t$);

- $(s \wedge s) = s$ (AND is neutral)

- $(s \wedge t) = (t \wedge s)$ (AND is commutative)
- $(s \vee s) = s$ (OR is neutral)
- $(s \vee t) = (t \vee s)$ (OR is commutative)

An allele is said to be dominant over another if it is expressed in heterozygosity. An allele is said to be recessive if it is only expressed in homozygosity. In the case of using the AND operator, allele 0 is dominant over allele 1, and vice versa for the OR operator.

Obviously, the dominance operator can vary from locus to locus. In the case that the allele set is larger than $\{0, 1\}$, similar dominance relations between alleles can be assigned. However, such dominance relations are not required for the scope of this text.

Coefficient of dominance. Instead of strictly choosing a dominant and recessive allele, the concept of a dominance coefficient can be used to introduce probabilities of dominance and recessiveness. The coefficient of dominance, symbolized by h, is the probability that a recessive allele is dominant in the case of heterozygosity. Note that, if a dominance coefficient is assumed, δ becomes a probabilistic function. As an example, if allele 1 is dominant over 0 with dominance coefficient h, then

$$\delta(\{0,1\}) = \begin{cases} 0 & \text{with probability } h \\ 1 & \text{with probability } 1-h. \end{cases}$$

In the OR dominance scheme with dominance coefficient h, the heterozygous genotype $\{0, 1\}$ has phenotype 0 with probability h, and phenotype 1 with probability $1-h$. The introduction of a dominance coefficient does not affect neutrality and symmetry properties (3.1) and (3.2).

Other dominance mappings. Other dominance relationships for diploid GAs, such as dominance determining loci and dominance masks, have been described in (Bidwell 1996; Collingwood, Corne, & Ross 1996; Greene 1999; Hadad & Eick 1997; Lewis, Hart, & Ritchie 1998; Liekens, ten Eikelder, & Hilbers 2003e; Smith & Goldberg 1992). Many variations of dominance can be observed in nature, and can be used in diploid GAs. Note that some of these implementations do not necessarily assume that a diploid individual is constructed as a multiset, and that neutrality and commutativity of the dominance schemes in these implementations may be missing.

3.3 Population Space

Let P denote a population, or multiset of individuals in Ω, and define p_i as the proportion of individuals $i \in \Omega$ in population P, hence $\sum_{i\in\Omega} p_i = 1$ and all p_i are positive reals. For a search space Ω, let the simplex Λ denote the set of all possible populations P, with Λ the set of stochastic vectors

$$\Lambda = \left\{ (p_0, \ldots, p_{|\Omega|-1})^\top \mid \sum_{i=0}^{|\Omega|-1} p_i = 1, p_i \geq 0 \right\}.$$

From now on, we assume that each population P, a multiset over Ω, corresponds with, and can be identified by a population vector $\mathbf{p} = \left(p_0, \ldots, p_{|\Omega|-1}\right)^\top \in \Lambda$. With a given multiset P, we identify a population vector $\mathbf{p}$. Let $r = |P|$ denote the population size of P. We let $P_i = rp_i$ denote the number of individuals with genome i.

For example, consider a population $P = \{0, 2, 2, 2, 3, 3\}$ of size $r = 6$ over $\Omega = \{0, 1, 2, 3\}$. This population P has $P_2 = 3$ individuals with genome 2, or proportion $p_2 = 1/2$. The corresponding population vector $\mathbf{p}$ equals $\mathbf{p} = (1/6, 0, 1/2, 1/3)^\top$. In the forthcoming equations, either p_i or P_i might be used, dependent on which of the two is best suited for the readability of the equations, but we always assume their relation by $P_i = rp_i$.

3.3.1 Infinite population size

A multiset or population P with infinite size over Ω is defined as a $|\Omega|$-dimensional vector $\mathbf{p}$, where each of the elements p_i corresponds to the proportion of individuals of type i in the population. All possible populations correspond to a unique point in the simplex Λ, and all points in Λ correspond to a unique infinitely large population P.

3.3.2 Finite population size

A finite population P is a multiset of cardinality $r \in \mathbb{N}$ over Ω. We say that r is the finite population size of P. Let $\pi \subset \Lambda$ denote the set of all stochastic population vectors $\mathbf{p}$ that correspond with a finite population P of size r over Ω. The set π of all possible states, or populations of the model, is a regular, discrete grid of states in simplex Λ of the infinite model, as depicted in Figure 3.1. All of the finite populations P correspond with a population vector $\mathbf{p} \in \Lambda$, but not all population vectors $\mathbf{p} \in \Lambda$ correspond with a finite population P of size r. The state space of the infinite model is thus continuous, where the state space of the finite model is discrete.

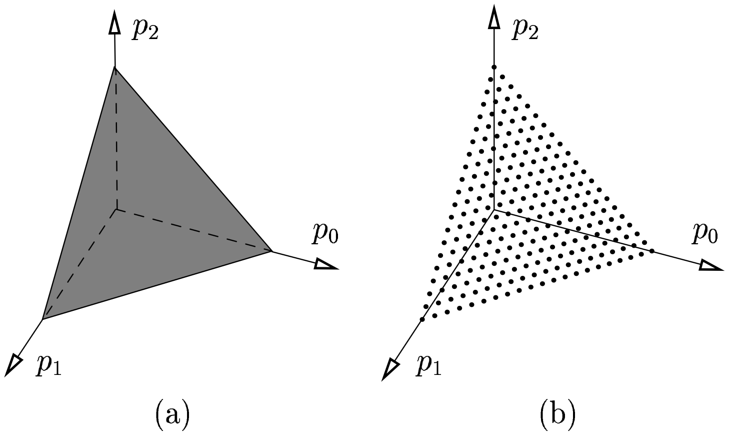


Figure 3.1: Three dimensional representation of (a) simplex Λ, state space of the infinite population model for $|\Omega| = 3$, and (b) π, state space of the finite population model for $|\Omega| = 3$ and $r = 20$.

Population space size. The number of possible multisets (populations) of size r over a set (search space) of size $n = |\Omega|$ is given by

$$|\pi| = \binom{r+n-1}{r} = \binom{r+n-1}{n-1}. \tag{3.3}$$

The number of possible populations grows polynomially with the size r of the population and size n of the search space. We can show this for r by finding a polynomial upper bound in r on the number of possible populations with

$$\begin{aligned} \binom{r+n-1}{r} &= \frac{(r+n-1)!}{r!(n-1)!} \\ &= \frac{1}{(n-1)!}\prod_{i=r+1}^{r+n-1} i \\ &\leq \frac{1}{(n-1)!}\prod_{i=r+1}^{r+n-1} r+n-1 \\ &= \frac{(r+n-1)^{(n-1)}}{(n-1)!}. \end{aligned} \tag{3.4}$$

Similarly, we can find a polynomial upper bound in n for $|\pi|$, namely

$$\binom{r+n-1}{r} \leq \frac{(r+n-1)^r}{r!}.$$

We want to point out that many authors in the genetic algorithms community wrongly assume that this growth is exponentially.

Although polynomially bounded, the number of possible populations for a reasonable population size or search space easily exceeds our computational capabilities. Section 3.8 discusses the numerical methods used for studying finite population models, and goes deeper into the problems faced when large state spaces are encountered.

Populations can be identified with a number i, $0 \leq i < |\pi|$. Appendix A details the enumeration and conversion of populations to their respective numbers.

3.4 Selection

Selection takes a population $P \in \pi$ and selects an individual from that population, based on a fitness function f. Individuals with a higher fitness have a higher probability of being selected from the population.

3.4.1 Fitness functions

Due to a wide range of applications in the later chapters of this thesis, we give an abstract definition of a fitness function that can be adapted later to fit the specific application. For the abstraction in this model, we assume that an individual's fitness is (obviously) dependent on the individual itself, the population P it resides in, and abstract environment parameters $\mathcal{E}$, other than P. Merely for notational reasons, we assume that population P is not part of this environment $\mathcal{E}$. Specific properties of fitness environments can be embedded into this environment, thus providing a common and versatile notation in the discussion of the future chapters' applications. Examples of environment parameters can be the current generation number in a time dependent fitness function, or another population for the modeling of co-evolution of populations.

We let $f(i, P, \mathcal{E})$ denote the fitness of an individual $i \in \Omega_S$ in its population P, given an environment $\mathcal{E}$. We assume that fitness maps to the set of nonnegative reals denoted by $\mathbb{R}^+$. In applications where the fitness function is static or only dependent on the population, but independent of other environment parameters, we neglect this parameter in our notation, i.e., $f(i, P) = f(i, P, \varnothing)$.

Mapping of search spaces. The fitness function f is defined over individuals in Ω_S, the search space of the optimization problem. In order to be able to determine the fitness of haploid and diploid individuals, the mapping from individuals in Ω_H and Ω_D to Ω_S as discussed previously is exploited.

For determining the fitness $f_H(i, P, \mathcal{E})$ of haploid individuals $i \in \Omega_H$, the corresponding element $i \in \Omega_S$ is used to determine the individual's fitness, i.e.,

$$f_H(i, P, \mathcal{E}) = f(i, P, \mathcal{E})$$

with $i \in \Omega_H = \Omega_S$ and P denoting a population of haploid individuals.

For determining the fitness of diploid individuals $i \in \Omega_D$, a dominance operator is used to determine the diploid individual's fitness. We can thus translate a fitness function f for elements in Ω_S to a fitness function f_D for diploid multisets in a diploid population P through the identification

$$f_D(i, P, \mathcal{E}) = f(\delta(i), \delta(P), \mathcal{E}).$$

This enables us to select diploid individuals according to fitness function f_D, which is related to the original fitness function f through the dominance mapping operator δ. Note that we also use the convention that the mapping operator δ is also defined for diploid populations, by mapping all individuals of this population using dominance operator δ, with $\delta(P) = \{\delta(i) \mid i \in P\}$.

For generality and ease of notation, the mapping of individuals to elements of the optimization problem's search space is implicitly used, thus adopting notation $f(i, P, \mathcal{E})$ not only for elements i of Ω_S, but also for the fitness of individuals of in Ω_H and Ω_D.

Mean fitness. The mean fitness $\overline{f}(P, \mathcal{E})$ of the individuals in population P (with corresponding stochastic population vector $\mathbf{p}$) over Ω, given their environment $\mathcal{E}$, equals

$$\overline{f}(P, \mathcal{E}) = \sum_{i \in \Omega} f(i, P, \mathcal{E}) p_i. \tag{3.5}$$

3.4.2 Fitness proportional selection

According to fitness function f, which maps elements of the search space Ω to the nonnegative reals $\mathbb{R}^+$, we can select an individual $i \in \Omega$ from population P with selection probability $S(i, P, \mathcal{E})$ where

$$\begin{aligned} S(i, P, \mathcal{E}) &= \mathbb{P}[i \text{ is selected from } P \text{ in environment } \mathcal{E}] \\ &= \frac{f(i, P, \mathcal{E}) p_i}{\sum_{i \in \Omega} f(i, P, \mathcal{E}) p_i} \\ &= \frac{f(i, P, \mathcal{E}) p_i}{\overline{f}(P, \mathcal{E})}. \end{aligned} \tag{3.6}$$

Note that $\sum_{i \in \Omega} S(i, P, \mathcal{E}) = 1$ holds. In the case that all individuals $i \in \Omega$ have fitness 0, and the population thus has a mean fitness 0, all individuals are selected with a probability independent of their fitness, i.e., with $S(i, P, \mathcal{E}) = p_i$. The fitness proportional selection method thus renders selected genotypes proportional to their fitness and abundance in population P.

Note that we assume that a higher fitness implies a higher selection probability, such that selection for minimization problems must be adapted accordingly. All of the optimization problems and their fitness functions studied in this text are maximization problems, such that the definition of fitness proportional selection is sufficient.

Other selection schemes, such as tournament or ranking selection, are discussed in Vose (1999b), but are out of scope for the text.

3.5 Genetic Operators

3.5.1 Mutation

Bit strings. Under the assumption that mutation is a bitwise operator, each bit in the bit string has a small probability μ to undergo a "bit flip". We can write the probability that a given bit string $s \in \Omega_S$ of length l mutates to become another bit string $t \in \Omega_S$ as a product of the mutation probability for each independent bit, by

$$\mathbb{P}\left[M_l(s) = t\right] = \prod_{i=0}^{l-1} \mu^{[s_i \neq t_i]}(1-\mu)^{[s_i = t_i]}. \tag{3.7}$$

Square bracketed notation $[e]$ represents 1 if boolean expression e is true, and 0 otherwise. If $d(s,t)$ denotes the Hamming distance between bit strings s and t, the probability that s mutates to t is equal to

$$\mathbb{P}\left[M_l(s) = t\right] = \mu^{d(s,t)}(1-\mu)^{l-d(s,t)}.$$

Strings. We can generalize the definition of locus-wise mutation for bit strings to strings where each of the loci i is assigned a different allele set Σ_i of size $|\Sigma_i|$. An allele at position i is mutated to a different allele with probability $(|\Sigma_i|-1)\mu$ and remains the same with probability $1-(|\Sigma_i|-1)\mu$. The probability that a given string s is mutated to t then becomes

$$\mathbb{P}\left[M_l(s) = t\right] = \prod_{i=0}^{l-1} \mu^{[s_i \neq t_i]}\left(1-\left(|\Sigma_i|-1\right)\mu\right)^{[s_i = t_i]} \tag{3.8}$$

which is a more general form of (3.7).

3.5.2 Crossover

Given are two "parent" strings $s, t \in \Omega_S$ of length l. We define $\mathbb{P}\left[X_l(s,t) = u\right]$ as the probability that crossing over of these strings results in string $u \in \Omega_S$. This probability depends on the crossover function used to create new children.

In the case of uniform crossover, the alleles at a locus of the resulting string u have a probability 1/2 of being copied from parent s and a probability of 1/2 of being copied from string t. The probability of s and t generating string u through uniform crossover is equal to

$$\mathbb{P}\left[X_l(s,t)=u\right]=\prod_{i=0}^{l-1}\left(\frac{1}{2}[u_i=s_i]+\frac{1}{2}[u_i=t_i]\right).$$

We assume this type of crossover throughout this text. For other crossover operators, such as a 1-point or n-point crossover types, see e.g., Vose (1999b).

3.5.3 Distributivity of crossover and mutation

The following theorem shows that the order of locus-wise mutation and uniform crossover of strings with length l is distributive. This theorem is a special case of Theorem 4.2 in (Vose 1999b).

Theorem 3.1. *If, for bit strings, mutation M_l is bit-wise and crossover X_l is uniform, then*

$$\forall s,t,u\in\Omega_S:\mathbb{P}\left[M_l(X_l(s,t))=u\right]=\mathbb{P}\left[X_l(M_l(s),M_l(t))=u\right]. \tag{3.9}$$

Proof. Since uniform crossover X_l is assumed and M_l is locus-wise, we can rewrite (3.9) for any $s,t,u\in\Omega_S$ as

$$\prod_{0\le i<l}\mathbb{P}\left[M_1(X_1(s_i,t_i))=u_i\right]=\prod_{0\le i<l}\mathbb{P}\left[X_1(M_1(s_i),M_1(t_i))=u_i\right]. \tag{3.10}$$

For each ith allele s_i, t_i, u_i of strings u,v and t, we prove that the order of mutation and crossover for one locus is distributive, i.e., that for $0\le i<l$

$$\mathbb{P}\left[M_1(X_1(s_i,t_i))=u_i\right]=\mathbb{P}\left[X_1(M_1(s_i),M_1(t_i))=u_i\right]. \tag{3.11}$$

This result can then be inserted into (3.10) to prove the theorem. We use notation $\neg s_i$ to denote $1-s_i$.

The left hand side of (3.11) is expanded as

$$\begin{aligned}
&\mathbb{P}\left[M_1(X_1(s_i,t_i))=u_i\right]\\
={}&\sum_{p=u_i,\neg u_i}\mathbb{P}\left[X_1(s_i,t_i)=p\right]\mathbb{P}\left[M_1(p)=u_i\right]\\
={}&\mathbb{P}\left[X_1(s_i,t_i)=u_i\right]\mathbb{P}\left[M_1(u_i)=u_i\right]+\mathbb{P}\left[X_1(s_i,t_i)=\neg u_i\right]\mathbb{P}\left[M_1(\neg u_i)=u_i\right]\\
={}&(\frac{1}{2}[s_i=u_i]+\frac{1}{2}[t_i=u_i])(1-\mu)+(\frac{1}{2}[s_i\neq u_i]+\frac{1}{2}[t_i\neq u_i])\mu,
\end{aligned}$$

$$\tag{3.12}$$

and the right hand side can be expanded as follows:

$$\begin{aligned}
&\mathbb{P}\left[X_1(M_1(s_i), M_1(t_i)) = u_i\right] \\
= &\sum_{p=s_i,\neg s_i} \sum_{q=t_i,\neg t_i} \mathbb{P}\left[X_1(p,q) = u_i\right] \mathbb{P}\left[M_1(s_i) = p\right] \mathbb{P}\left[M_1(t_i) = q\right] \\
= &\ \mathbb{P}\left[X_1(s_i,t_i) = u_i\right] \mathbb{P}\left[M_1(s_i) = s_i\right] \mathbb{P}\left[M_1(t_i) = t_i\right] + \\
&\ \mathbb{P}\left[X_1(s_i,\neg t_i) = u_i\right] \mathbb{P}\left[M_1(s_i) = s_i\right] \mathbb{P}\left[M_1(t_i) = \neg t_i\right] + \\
&\ \mathbb{P}\left[X_1(\neg s_i,t_i) = u_i\right] \mathbb{P}\left[M_1(s_i) = \neg s_i\right] \mathbb{P}\left[M_1(t_i) = t_i\right] + \\
&\ \mathbb{P}\left[X_1(\neg s_i,\neg t_i) = u_i\right] \mathbb{P}\left[M_1(s_i) = \neg s_i\right] \mathbb{P}\left[M_1(t_i) = \neg t_i\right] \\
= &\left(\frac{1}{2}[s_i = u_i] + \frac{1}{2}[t_i = u_i]\right)(1-\mu)^2 + \\
&\left(\frac{1}{2}[s_i = u_i] + \frac{1}{2}[t_i \neq u_i]\right)\mu\,(1-\mu) + \\
&\left(\frac{1}{2}[s_i \neq u_i] + \frac{1}{2}[t_i = u_i]\right)\mu\,(1-\mu) + \\
&\left(\frac{1}{2}[s_i \neq u_i] + \frac{1}{2}[t_i \neq u_i]\right)\mu^2
\end{aligned}\tag{3.13}$$

We now make a distinction between two separate cases, i.e., $s_i \neq t_i$ and $s_i = t_i$.

- $s_i \neq t_i$

  In this case we can fill in (3.12) and (3.13) as follows:
  $\mathbb{P}\left[M_1(X_1(s_i,t_i)) = u_i\right] = \frac{1}{2}(1-\mu) + \frac{1}{2}\mu = \frac{1}{2}$ and
  $\mathbb{P}\left[X_1(M_1(s_i), M_1(t_i)) = u_i\right] = \frac{1}{2}(1-\mu)^2 + \mu(1-\mu) + \frac{1}{2}\mu^2 = \frac{(1-\mu+\mu)^2}{2} = \frac{1}{2}$
  This shows that (3.11) holds if $s_i \neq t_i$.

- $s_i = t_i$

  We make an extra distinction between $s_i = t_i = u_i$ and $s_i = t_i \neq u_i$:

  - $s_i = t_i = u_i$

    We can again fill in (3.12) and (3.13):
    $\mathbb{P}\left[M_1(X_1(s_i,t_i)) = u_i\right] = 1 - \mu$ and
    $\mathbb{P}\left[X_1(M_1(s_i), M_1(t_i)) = u_i\right] = (1-\mu)^2 + (1-\mu)\mu = 1 - \mu,$
    thus (3.11) holds in this case.

- $s_i = t_i \neq u_i$
  We fill this also in (3.12) and (3.13):
  $\mathbb{P}\left[M_1(X_1(s_i, t_i)) = u_i\right] = \mu$ and
  $\mathbb{P}\left[X_1(M_1(s_i) = M_1(t_i)) = u_i\right] = (1-\mu)\mu + \mu^2 = \mu$,
  thus (3.11) holds in this case.

The fact that (3.11) holds for all possible combinations of s_i, t_i, u_i concludes the proof of our theorem.

□

We have shown the theorem in the case of bit strings, but it can easily be extended for general strings by replacing all occurrences of mutation rate μ for bits with the mutation rate $(|\Sigma_i| - 1)\,\mu$ for a locus i with $|\Sigma_i|$ allele types. Notation $\neg s_i$ is then used to denote any element in $\Sigma_i \setminus \{s_i\}$, i.e., the alleles in allele set Σ_i that are not s_i.

3.5.4 Meiosis

Meiosis is a stochastic function $\theta : \Omega_D \rightarrow \Omega_H^d$ that combines crossover and mutation and produces d haploid daughter gametes from a diploid zygote. In nature, meiosis produces $d = 4$ daughter gametes from one cell. This number is due to the duplication of the chromosomes, followed by two consecutive splits of the cell during meiosis. In genetic algorithms and models of natural evolution, it is commonly assumed that crossover and mutation of two haploid parents result in one child individual. We will assume $d = 1$ and therefore $\theta : \Omega_D \rightarrow \Omega_H$.

Given an individual $\{s, t\} \in \Omega_D$, meiosis produces a gamete $u \in \Omega_H$ with probability

$$\mathbb{P}\left[\theta(\{s,t\}) = u\right] = \mathbb{P}\left[X_l(M_l(s), M_l(t)) = u\right],$$

based on the probabilities for mutation and crossover as studied before. Note that because of Theorem 3.1, this probability can also be written as

$$\mathbb{P}\left[\theta(\{s,t\}) = u\right] = \mathbb{P}\left[M_l(X_l(s,t)) = u\right]. \tag{3.14}$$

This latter ordering of mutation and crossover is more cost effective if computational time has to be taken into account when implementing a meiotic operator in a genetic algorithm.

3.5.5 Haploid and diploid reproduction

According to the processes in haplontic and diplontic life cycles, as discussed in Section 2.3.2, we can now combine our definitions of selection and reproduction to compute the probabilities of an individual being generated from a given parent population.

Reproduction of haploid individuals. The haplontic dogma in the study and implementation of genetic algorithms is based on the processes of natural haplontic life cycles. In natural haploid populations, two parents' genomes are fused through fertilization to become a diploid individual that undergoes meiosis, where 4 haploid daughter individuals are created. In genetic algorithms, selected parents are recombined using crossover and mutation operators to form new child individuals for the next generation. This is essentially the same interpretation of the haplontic life cycle in natural populations, but with different numbers of resulting daughter individuals. By using meiosis operator θ, we can model the generation of new individuals from a haploid parent population in our formal framework.

A haploid individual $i \in \Omega_H$ is generated from a haploid population P_H with probability

$$G\left(i, P_H, \mathcal{E}\right) = \sum_{j,k \in \Omega_H} \mathbb{P}\left[\theta\left(\{j,k\}\right) = i\right] \cdot S\left(j, P_H, \mathcal{E}\right) \cdot S\left(k, P_H, \mathcal{E}\right) \tag{3.15}$$

where $S\left(x, P_H, \mathcal{E}\right)$ denotes the probability that an individual with genome $x \in \Omega_H$ is selected from population P_H, in an environment $\mathcal{E}$, as defined in (3.6).

Reproduction of diploid individuals. In diploid populations with diplontic life cycles, meiosis takes place in selected individuals, and the resulting haploid gametes are fused through fertilization to become new diploid individuals. The implementation of a diploid genetic algorithm is also according to this scheme. The homologous strings that make up a diploid individual are recombined through crossover and mutation. Gametes, or haploid strings, generated with these operators from selected parents are then fused to form a new diploid individual for the next generation. By reusing the definitions of selection and meiosis operator θ, we can easily formalize the process of generating new individuals for our model of populations with diplontic life cycles.

A diploid individual $\{i,j\} \in \Omega_D$ is generated from a diploid population P_D with probability

$$\begin{aligned} G\left(\{i,j\}, P_D, \mathcal{E}\right) = \sum_{\{k,l\},\{m,n\} \in \Omega_D} & \left(\mathbb{P}\left[\theta\left(\{k,l\}\right) = i\right] \cdot \mathbb{P}\left[\theta\left(\{m,n\}\right) = j\right] + \right. \\ & \left. \mathbb{P}\left[\theta\left(\{k,l\}\right) = j\right] \cdot \mathbb{P}\left[\theta\left(\{m,n\}\right) = i\right]\right) \cdot \\ & S\left(\{k,l\}, P_D, \mathcal{E}\right) \cdot S\left(\{m,n\}, P_D, \mathcal{E}\right). \end{aligned} \tag{3.16}$$

Computational equivalence of haploidy and diploidy. In Liekens et al (2003e), we have demonstrated how a diploid SGA can be transformed into a haploid SGA with equivalent behavior, and vice versa. We have shown that haploid reproduction can be simulated by diploid individuals, by constructing specific representations of haploid individuals in diploid genomes, and by implementing specific crossover and mutation operators to simulate the haploid reproduction scheme in diploid individuals, and vice versa.

Consequently, if a diploid SGA is being implemented, there are two possible ways to design a representation of the genotype and its genetic operators. The diploid genetic algorithm can be built according to the definition of the diploid simple genetic algorithm. Secondly, the simulation of the diploid SGA inside a haploid SGA can also be adopted. If these implementations follow the construction guidelines given, the resulting behavior of both implementations is the same. As a result, theorems that apply to genetic algorithms that allow non-specific limitations to the genetic operators, such as convergence theorems of genetic algorithms (Eiben, Aarts, & Van Hee 1992), also apply to the diploid simple genetic algorithm. The construction showing this equivalence is, however, out of scope for this text.

3.6 Constructing New Populations

An instantiation of a SGA is a sequence of random events (in this case, populations) in time

$$P(0) \to P(1) \to P(2) \to P(3) \to \dots \tag{3.17}$$

which is governed by a *transition rule* τ. A population $P(g)$ at time step g is mapped to a new population $P(g+1)$ at time step $g+1$ with probability $\mathbb{P}\left[\tau(P(g)) = P(g+1)\right]$. We now characterize the transitional mapping rule τ for the SGA, based on the probabilities of that an individual i is generated through selection and reproduction according to $G(i, P, \mathcal{E})$.

Let $\mathcal{G} : \Lambda \to \Lambda$ denote the *heuristic function* that maps a stochastic population vector $\mathbf{p}$ to a new vector $\mathbf{p}' = \mathcal{G}(\mathbf{p}) \in \Lambda$ in one time step. The heuristic function combines all elements of one generation. Element p'_i now denotes the probability that $i \in \Omega$ is chosen to be in the population at the next iteration of the algorithm. Any heuristic function $\mathcal{G} : \Lambda \to \Lambda$ can be a model of the discrete time dynamics of a population based algorithm. For the SGA in an environment $\mathcal{E}$, $\mathcal{G}$ is defined by

$$\mathcal{G} : \mathbf{p} \mapsto \mathbf{p}' \text{ with } \forall i \in \Omega : p'_i = \left(\mathcal{G}\left(\mathbf{p}\right)\right)_i = G(i, P, \mathcal{E}). \tag{3.18}$$

For a given population $\mathbf{p}$, $\mathcal{G}(\mathbf{p})$ thus gives the sampling distribution over the individual types from which r individuals are sampled to construct the population of size r at the next generation. In an SGA, the transition from one generation to the next is thus constructed by collecting r samples from $\mathcal{G}(\mathbf{p})$, where $\mathbf{p}$ represents the current population. Transition rule τ corresponds accordingly to the heuristic function $\mathcal{G}$.

3.6.1 Infinite populations

Let $\mathbf{p}$ represent an infinite population at one time step. If the distribution over individual types at the next generation $\mathbf{p}' = \mathcal{G}(\mathbf{p})$ is sampled an infinite number of times, then the resulting infinitely large population is uniquely defined by this distribution. Consequently, for infinitely large populations, transition rule τ is a delta function over Λ by

$$\mathbb{P}\left[\tau(P(g)) = P(g+1)\right] = \begin{cases} 1 & \text{if } \mathbf{p}(g+1) = \mathcal{G}(\mathbf{p}(g)) \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

In the case of infinitely large populations, the model of the SGA is deterministic, since the population after a generation is uniquely defined by the heuristic function $\mathcal{G}$. Hence, the sequence of random events describing the dynamics of an infinitely large population is deterministic, and uniquely defined for a given initial population $P(0)$, i.e., the sequence of random events

$$P(0) \to P(1) \to P(2) \to P(3) \to \ldots$$

is given by

$$P(0) \to \mathcal{G}^1(P(0)) \to \mathcal{G}^2(P(0)) \to \mathcal{G}^3(P(0)) \to \ldots$$

in the case of an infinitely large population size.

3.6.2 Finite populations

In the case of populations with a fixed and finite number r of individuals, we sample the results of reproduction r times to construct a new population for the next generation. This is according to our assumption of the generational process in an SGA, where a new non-overlapping population of r individuals is created at each generation. Along these lines, one may also construct a model of a GA where only a part of the population is replaced by new individuals.

With the introduction of finite population sizes, the process is no longer deterministic and becomes stochastic. The probability that population $P(g+1)$ with population size r is generated through sampling in one generation, based on heuristic $\mathcal{G}$, from population $P(g)$ is equal to

$$\mathbb{P}\left[\tau(P(g)) = P(g+1)\right] = \frac{r!}{\prod_{i\in\Omega}\left(P(g+1)\right)_i!}\prod_{i\in\Omega} G(i, P(g), \mathcal{E})^{(P(g+1))_i}. \tag{3.20}$$

Recall that $\left(P(g+1)\right)_i$ denotes the number of individuals with genome i in $P(g+1)$. The multinomial coefficient $\frac{r!}{\prod_{i\in\Omega}(P(g+1))_i!}$ computes the number of possible arrangements for a population $P(g+1)$ of size r. The other factors denote the probability that such an arranged population is sampled from the reproduction process.

Note that the finite model and the infinite model are closely related, since the finite model samples the infinite population model r times. The expected population in one time step of the finite model is equal to the next population in the corresponding infinite model.

3.7 Transient and Long Run Behavior

We now have all elements in place to model the simple genetic algorithm, and study its transient, i.e., short time, and limit behavior, both in the case of finitely and infinitely large populations. In this section, we discuss how the SGA is initialized and how the subsequent transient and limit behavior can be studied.

3.7.1 Representations

Because of the difference in stochasticity of the finite and infinite population model, we describe an infinite population model with proportions of individual types, and a finite population model with distributions over populations.

Infinite populations. An infinite population P is represented by a stochastic population vector $\mathbf{p} \in \Lambda$. Each of the entries p_i represents the proportion of individual $i \in \Omega$ residing in the population. Since the model for infinite populations is deterministic, we can represent its transient behavior as a discrete time chain of population vectors, governed by heuristic function $\mathcal{G}$.

Finite populations. In contrast to the deterministic infinite population model, the finite model is stochastic. The resulting model is a discrete time, finite Markov chain $\langle \pi, \tau \rangle$, over the state space of the populations π, with transition probability matrix T with entries given by transition rule τ, i.e.,

$$T_{P',P} = \mathbb{P}\left[\tau(P) = P'\right]. \tag{3.21}$$

Numbers of the rows and columns (or states) are given by the population numbers as discussed in Appendix A. We study the behavior of the finite population models with probability distributions over all possible population configurations, by iterating over consecutive generations, given an initial distribution over the states of the system. A distribution $\mathbf{x}$ is a $|\pi|$-dimensional vector, whose entries x_P denote the probability of residing in state P. Clearly, all distributions $\mathbf{x}$ are stochastic vectors with $\forall P \in \pi : x_P \geq 0$ and $\sum_{P \in \pi} x_P = 1$.

3.7.2 Measures of deviation

In order to study the influence of dropping the assumption of infinitely large populations, both the infinite and finite models are run side by side, with the same parameters. In order to quantify the differences among both models, we need a means to measure how the behavior in the finite model deviates from the expected behavior of the infinite model.

At any time step, the finite model is represented by a distribution $\mathbf{x}$ over a subset in population space Λ, where the infinite model's distribution is represented as a delta function, with probability 1 at a vector $\mathbf{p}_\infty$ in this population space Λ and 0 everywhere else. Note that notation $\mathbf{p}_\infty$ is used to denote an arbitrary infinite population, not the population after an infinite number of steps. Thus, an infinite population $\mathbf{p}_\infty$ and distribution $\mathbf{x}$ over finite populations in π are given, and we need a measure between these two distributions which tells us how the finite population distribution differs from the infinite population, with respect to their distance in set Λ. As the finite model's distribution's weights are allocated further away from the infinite population model's vector, their mutual distance becomes larger. We should point out that in the limit or steady state behavior, the expected behavior of the infinite population model may become a more complex distribution over the population space instead of a simple delta function. We'll regard this behavior as exceptional and introduce corrections to the method of measuring when required.

Since we want to measure how the behavior in the finite model *deviates* from the behavior in the infinite model, we need a measure that quantifies how the probabilities are relocated in between the two models' distributions. Common distance measures that quantify differences among distributions, such as the Cartesian distance between vectors, or the Kullback-Leibler divergence (Kullback & Leibler 1951; Kullback 1959; Cover & Thomas 1991; Qian 2001) do not give the required information, as they do not give any information on how probabilities are redistributed among states of the systems, with respect to the set on which the distributions are laid out. Secondly, these distance measures require the probability distributions to be discrete and be defined over the same domain. This is not the case for the finite and infinite distributions, as the finite populations are distributed over a grid which is a subset over Λ, where the peak of the delta function describing the infinite population model's distribution can be anywhere in Λ. An option could be to resample the infinite model's representation to the grid of the finite model, but this would result in a distance measure that is biased toward the population size that is used for resampling the infinite model's delta function.

The mean $\mathbb{E}[\mathbf{p}]$ of distribution $\mathbf{x}$ and variance $\mathbb{E}[\mathbf{p} - \mathbb{E}[\mathbf{p}]]^2$ about this mean

$$\mathbb{E}[\mathbf{p}] = \sum_{\mathbf{p} \in \pi} x_{\mathbf{p}} \mathbf{p} \text{ and} \tag{3.22}$$

$$\mathbb{E}\left[\mathbf{p}-\mathbb{E}\left[\mathbf{p}\right]\right]^2 = \sum_{\mathbf{p}\in\pi} x_{\mathbf{p}}\left(\mathbf{p}-\mathbb{E}\left[\mathbf{p}\right]\right)^2 \tag{3.23}$$

give an idea of how the finite model deviates from the infinite model. We let $(\mathbf{y})^2$ denote $\sum_{i\in\Omega} y_i^2$. Indeed, the distance between the mean $\mathbb{E}[\mathbf{p}]$ of the finite model's probability distribution (which is a population vector in Λ) and the population vector $\mathbf{p}_\infty$ of the infinite model shows how the average behavior of the finite population model differs from the infinite population model. The variance $\mathbb{E}\left[\mathbf{p}-\mathbb{E}\left[\mathbf{p}\right]\right]^2$ of the finite population model gives an indication of the variance $\mathbb{E}\left[\mathbf{p}-\mathbf{p}_\infty\right]^2$ of the distance between the distribution and the infinite population model's state.

We use a slightly modified version of the standard deviation of the finite population model, which also incorporates the distance between the mean and the infinite model. We measure the *expected deviation* $\sigma_{\mathbf{p}_\infty}(\mathbf{x})$ of the finite population model $\mathbf{x}$ with respect to the infinite population model $\mathbf{p}_\infty$, with

$$\sigma_{\mathbf{p}_\infty}(\mathbf{x}) = \sqrt{\mathbb{E}\left[\mathbf{p}-\mathbf{p}_\infty\right]^2} = \sqrt{\sum_{\mathbf{p}\in\pi} x_{\mathbf{p}}\left(\mathbf{p}-\mathbf{p}_\infty\right)^2} \tag{3.24}$$

where $\mathbf{p}$ represents the stochastic population vector in Λ that corresponds with population P.

Note that there indeed exists a close relationship between the central moments of distribution $\mathbf{x}$ and this proposed distance measure, with

$$\begin{aligned}
\sigma^2_{\mathbf{p}_\infty}(\mathbf{x}) &= \mathbb{E}\left[\mathbf{p}-\mathbf{p}_\infty\right]^2 \\
&= \sum_{\mathbf{p}\in\pi} x_{\mathbf{p}}\left(\mathbf{p}-\mathbf{p}_\infty\right)^2 \\
&= \sum_{\mathbf{p}\in\pi} x_{\mathbf{p}}\left(\mathbf{p}-\mathbb{E}[\mathbf{p}]+\mathbb{E}[\mathbf{p}]-\mathbf{p}_\infty\right)^2 \\
&= \sum_{\mathbf{p}\in\pi} x_{\mathbf{p}}\left(\mathbf{p}-\mathbb{E}[\mathbf{p}]\right)^2 + \\
&\quad 2\sum_{\mathbf{p}\in\pi} x_{\mathbf{p}}\left(\mathbf{p}-\mathbb{E}[\mathbf{p}]\right)\left(\mathbb{E}[\mathbf{p}]-\mathbf{p}_\infty\right) + \\
&\quad \sum_{\mathbf{p}\in\pi} x_{\mathbf{p}}\left(\mathbb{E}[\mathbf{p}]-\mathbf{p}_\infty\right)^2 \\
&= \sum_{\mathbf{p}\in\pi} x_{\mathbf{p}}\left(\mathbf{p}-\mathbb{E}[\mathbf{p}]\right)^2 + \\
&\quad 2\left(\mathbb{E}[\mathbf{p}]-\mathbf{p}_\infty\right)\sum_{\mathbf{p}\in\pi} x_{\mathbf{p}}\left(\mathbf{p}-\mathbb{E}[\mathbf{p}]\right) +
\end{aligned}$$

$$
\begin{aligned}
& (\mathbb{E}[\mathbf{p}] - \mathbf{p}_\infty)^2 \sum_{\mathbf{p}\in\pi} x_{\mathbf{p}} \\
&= \mathbb{E}[\mathbf{p} - \mathbb{E}[\mathbf{p}]]^2 + 2(\mathbb{E}[\mathbf{p}] - \mathbf{p}_\infty)\, 0 + (\mathbb{E}[\mathbf{p}] - \mathbf{p}_\infty)^2\, 1,
\end{aligned}
$$

thus

$$
\mathbb{E}[\mathbf{p} - \mathbf{p}_\infty]^2 = \mathbb{E}[\mathbf{p} - \mathbb{E}[\mathbf{p}]]^2 + (\mathbb{E}[\mathbf{p}] - \mathbf{p}_\infty)^2,
$$

i.e., the expected variance with respect to the infinite model's population $\mathbf{p}_\infty$ is the sum of the variance around the mean of distribution $\mathbf{x}$ and the squared distance between the mean of this distribution and the population vector of the infinite model. As such, the expected deviation combines the finite model's variance and mean. Of course, we lose information with respect to the original moments by compressing them into one number, but the expected deviation achieves the descriptive value we needed for the measure of distance between the two given distributions. The proposed measure now gives an idea of how the finite population model deviates from the infinite population's prediction, on average.

Relation to genetic drift. In Chapter 2, we have defined genetic drift as the effect of random sampling in finite populations, which causes frequencies of alleles to diverge from the expected deterministic dynamics of a corresponding infinite population model. As the expected deviation measures the difference between a deterministic infinite population model on one hand, and a stochastic model of finite populations, it allows us to quantify the amount of genetic drift when dropping the assumption of an infinitely large population. Indeed, as we run an infinite and finite model side by side, with the same selective pressure and reproductional parameters, and measure the expected deviation of the finite model with respect to the infinite model at each time step, we can observe how genetic drift becomes apparent over time in the finite population model.

Similarly, we can analyze the influences of the selective and reproductive parameters and a varying population size on genetic drift in the long term. Measuring the expected deviation of the limit behavior of models with differing parameters with respect to their corresponding infinite population models, allows us to study how changes in selective pressure, mutation rate and population size reflect on the amount of genetic drift for the given parameters.

Expected deviation of population properties. The discussion of expected deviation of a distribution over finite models with respect to an infinite model has, until now, only focused on the distance in between populations in Λ. Other distances, measuring the differences with respect to other properties of the populations, may also be of interest.

A measure of expected deviation in which we are also interested, is the expected deviation of the fitness distribution of a finite population model, with respect to the fitness in an infinitely large population. Such a measure shows how the performance of a population is affected by dropping the assumption of infinitely large populations. The expected deviation of the mean population fitness of a distribution $\mathbf{x}$ over π with respect to the mean fitness of an infinitely large population $\mathbf{p}_\infty \in \Lambda$, in an environment $\mathcal{E}$ is equal to

$$\sqrt{\sum_{\mathbf{p}\in\pi} x_\mathbf{p}\left(\overline{f}(\mathbf{p},\mathcal{E}) - \overline{f}(\mathbf{p}_\infty,\mathcal{E})\right)^2}. \tag{3.25}$$

In this case, $\overline{f}$ is adopted as a mapping of the population vectors in Λ to a different space (in this case $\mathbb{R}^+$, the one dimensional space of mean population fitness). Other mappings, that extract specific properties from the populations, can also be used to study the deviation of properties between a finite and infinite model.

3.7.3 Initial population

Infinite populations. The SGA with an infinitely large population is started with an initial population $\mathbf{p}(0)$. Upon the construction of an initial population for the SGA, we assume that each of the elements in Ω_S has an equal chance of being assigned to a genome in an individual.

Since the search space of haploid genomes equals the search space of the optimization problem, with $\Omega_H = \Omega_S$, the initial population for studying the dynamics is assumed to be represented with

$$\forall s \in \Omega_H : (\mathbf{p}(0))_s = \frac{1}{|\Omega_S|} \tag{3.26}$$

or

$$\mathbf{p}(0) = \frac{\mathbf{1}}{|\Omega_S|}$$

where $\mathbf{1}$ represents the vector of size $|\Omega_S|$ of all 1s.

A diploid genome in Ω_D is a multiset of two elements from search space Ω_S. If elements in Ω_S are distributed uniformly among the two elements in a diploid individual, then diploid individuals $\{s,t\}$ with $s \neq t$ have twice the probability to appear in the population compared to the diploid individuals with a genome $\{s,s\}$. Otherwise stated, drawing the two elements for compositing a diploid individual which is a multiset causes these drawings not to be independent. The initial population for a diploid system is thus assumed to be represented with

$$\forall \{s,t\} \in \Omega_D : (\mathbf{p}(0))_{\{s,t\}} = \begin{cases} \frac{1}{|\Omega_S|^2} & \text{if } s = t \\ 2\frac{1}{|\Omega_S|^2} & \text{if } s \neq t. \end{cases} \tag{3.27}$$

Finite populations. If the probability q_i is given that a genome $i \in \Omega$ is chosen to be in the initial population of the SGA, then the probability that a population P resides in the initial distribution $\mathbf{x}(0)$ of the genetic algorithm is given by the multinomial distribution, i.e.,

$$(\mathbf{x}(0))_P = \frac{r!}{\prod_{i\in\Omega} P_i!} \prod_{i\in\Omega} q_i^{P_i}. \tag{3.28}$$

The distribution gives the probability that a haploid population is sampled, with r samples, from $\mathbf{q}(0)$ which is defined as $\mathbf{p}(0)$ in (3.26). This is similar to the sampling process in the transition rule τ of the SGA in (3.20). For the initial distribution $\mathbf{x}(0)$ over the states of the haploid model, we assume that each genome in Ω_H has an equal probability of being present in the population, similar to the initialization of infinitely large haploid populations, as in (3.26). Since the probability q_i for a haploid individual to be in the initial population is the same for all haploid genomes $s \in \Omega_H$, i.e., $q_s = 1/|\Omega_H|$, we can rewrite the initial distribution (3.28) for the haploid model with

$$\begin{aligned}(\mathbf{x}(0))_P &= \frac{r!}{\prod_{s\in\Omega_H} P_s!} \prod_{s\in\Omega_H} |\Omega_H|^{-P_s} \\ &= \frac{r!}{\prod_{s\in\Omega_H} P_s!} |\Omega_H|^{-r}.\end{aligned}$$

The formulation of the initial distribution over diploid populations cannot be simplified in a similar manner.

Example 3.1 (Initial haploid distribution, 1 locus, 2 alleles). As an example, consider the haploid model with search space $\Omega_H = \Omega_S = \{0, 1\}$, i.e., the haploid genomes with one locus and 2 alleles. The initial infinitely large population for this model is $\mathbf{p}_\infty(0) = (1/2, 1/2)^\top$. The corresponding initial distribution over the states of the finite model, denoted by π_H, with population size r is given by the binomial distribution $\mathbf{x}_H$ with

$$(\mathbf{x}_H(0))_P = \frac{r!}{P_0!P_1!} 2^{-r}. \tag{3.29}$$

Since the mean $\mathbb{E}[\mathbf{p}]$ of this distribution is equal to the initial population vector $\mathbf{p}_\infty$ of the infinite population model, the expected deviation $\sqrt{\mathbb{E}[\mathbf{p} - \mathbf{p}_\infty]^2}$ of the initial distribution of the infinite model with respect to the initial infinite population is equal to the standard deviation $\sqrt{\mathbb{E}[\mathbf{p} - \mathbb{E}[\mathbf{p}]]^2}$ of the finite model's distribution around its mean $\mathbb{E}[\mathbf{p}]$, which is equal to $\sqrt{rp_{\infty,0}p_{\infty,1}} = \frac{\sqrt{r}}{2}$ (Papoulis & Unnikrishna 2002).

The initial distributions of these models for $r = 10$ and $r = 30$ are depicted in Figure 3.2.

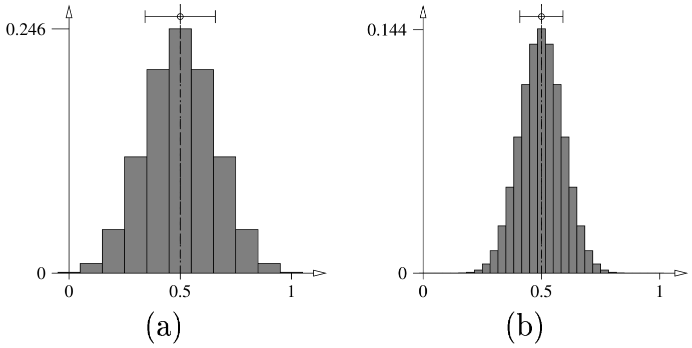


Figure 3.2: Initial distributions of haploid model with 1 locus, 2 alleles for (a) $r = 10$ and (b) $r = 30$. The horizontal axis represents the proportion of 0 alleles in the population. The vertical axis represents the probability of residing in the given state. The vertical dashed line represents the initial infinite population, the error bars denote the expected deviation of the finite population model with respect to the infinite population model.

Example 3.2 (Initial diploid distribution, 1 locus, 2 alleles). In similarity with the previous example, consider diploid individuals with one locus and 2 alleles. We let π_D denote the search space of finite diploid populations with genomes in $\Omega_D = \{\{0,0\},\{0,1\},\{1,1\}\}$. If we use the ordering of genome types as in the latter set, the initial infinitely large population for this model is $\mathbf{p}_\infty(0) = (1/4, 1/2, 1/4)^\top$. The corresponding initial distribution over the states of the finite model, with population size r is given by the multinomial distribution $\mathbf{x}_D$ with for all $P \in \pi_D$

$$(\mathbf{x}_D(0))_P = \frac{r!}{P_{\{0,0\}}!P_{\{0,1\}}!P_{\{1,1\}}!} 4^{-P_{\{0,0\}}} 2^{-P_{\{0,1\}}} 4^{-P_{\{1,1\}}}. \tag{3.30}$$

We assume a dominance mapping δ according to the binary operator $\wedge$, i.e., we assume that allele 0 is dominant over recessive allele 1. We can now phenotypically map distribution $\mathbf{x}_D$ over π_D to a distribution $\delta(\mathbf{x}_D)$ over π_H representing the distribution as it is expressed in the selection process with

$$\left(\delta\left(\mathbf{x}_D\right)\right)_Q = \sum_{P \in \pi_D} \begin{cases} (\mathbf{x}_D)_P & \text{if } Q = \delta(P) \\ 0 & \text{otherwise} \end{cases} \tag{3.31}$$

for all $Q \in \pi_H$. Figure 3.3 depicts the initial probability distributions of the diploid model, after these have been mapped phenotypically to a distribution over the possible representative haploid populations, using the above construction, for $r = 10$ and $r = 30$. Note that these phenotypically mapped distributions are equal to the distributions of the haploid population according to an initial infinite population $(3/4, 1/4)^\top$. Indeed, $p_{\infty,\{0,0\}} + p_{\infty,\{0,1\}} = 1/4 + 1/2$ of the initial individuals (those with diploid genotype $\{0,0\}$ and $\{0,1\}$) are expressed

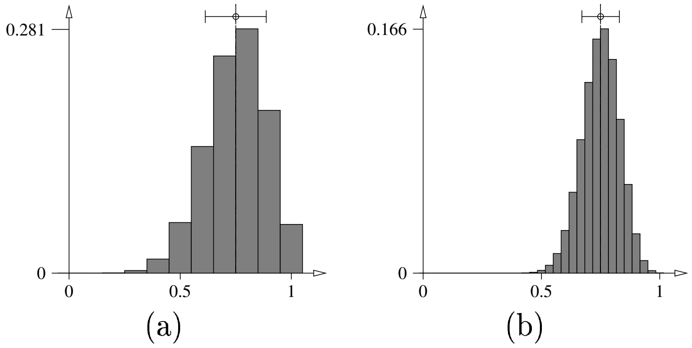


Figure 3.3: Initial phenotypically mapped distribution over populations for the diploid model with 1 locus, 2 alleles for (a) $r = 10$ and (b) $r = 30$. Allele 0 is dominant over allele 1. The horizontal axis represents the proportion of phenotype 0 in the populations.

phenotypically as 0, where the other quarter $(p_{\infty,\{1,1\}})$ of the initial population expresses as 1 (individuals with genome $\{1,1\}$).

In distribution $\delta(\mathbf{x}_D)$, the mean of the distribution is at $(3/4, 1/4)^\top$ and the standard or expected deviation is equal to $\sqrt{r(p_{\infty,\{0,0\}} + p_{\infty,\{0,1\}})p_{\infty,\{1,1\}}} = \frac{\sqrt{3}}{2}\frac{\sqrt{r}}{2}$, i.e., the initial expected deviation (after phenotypic mapping) is smaller than the expected deviation of the haploid model, with a factor $\frac{\sqrt{3}}{2}$.

3.7.4 Transient behavior

Infinite populations. As the heuristic function $\mathcal{G}$ is deterministic for infinite populations, consecutive populations $\mathbf{p}(g)$ are computed by iterating $\mathcal{G}$, i.e., $\mathbf{p}(g)$ for $g > 0$ is determined with

$$\mathbf{p}(g) = \mathcal{G}\left(\mathbf{p}(g-1)\right) \text{ or } \mathbf{p}(g) = \mathcal{G}^g\left(\mathbf{p}(0)\right).$$

This iterative process of heuristic $\mathcal{G}$ can then be used to study the transient behavior of the SGA with an infinite population size.

Finite populations. When the Markov chain is initialized with a distribution $\mathbf{x}(0)$ over the states, we can determine the distribution over the states at generation $g > 0$ with

$$\mathbf{x}(g) = T\mathbf{x}(g-1) \text{ or } \mathbf{x}(g) = T^g\mathbf{x}(0).$$

The consecutive distributions of the model can then be used to study the transient behavior of the SGA, when it is initialized according to distribution $\mathbf{x}(0)$.

3.7.5 Limit behavior

Infinite populations. The fixed points $\hat{P}$ of the heuristic that governs the infinite population model, with $\hat{p}_i = G(i, \hat{P}, \mathcal{E})$, or $\hat{\mathbf{p}} = \mathcal{G}\left(\hat{\mathbf{p}}\right)$ can be computed, under the assumption that they exist. A fixed point $\hat{\mathbf{p}}$ of $\mathcal{G}$ is stable if the eigenvalues λ of the Jacobian of $\mathcal{G}$ evaluated at $\hat{\mathbf{p}}$ are according to $|\lambda| < 1$. The fixed points and their stability properties can be adopted to investigate the long term behavior of the evolutionary system under selection and variation. The discovery of attracting fixed points of $\mathcal{G}$ in Λ is often sufficient to describe the limit behavior of the evolutionary systems.

Vose (1999b) gives an overview of methods to find the fixed points of the dynamical systems. It is noted that the direct computation of fixed points of the heuristic is often computationally hard and laborious. Because of the complexity in the construction of the heuristic, and commonly large number of variables involved, obtaining closed formula solutions for the fixed points is often hard or computationally expensive. A successful method used commonly throughout this thesis is the use of iteration of the infinitely large model, with an appropriately chosen initial population. This method is feasible, as the evolutionary process governed by $\mathcal{G}$ is usually focused toward states in Λ that correspond to populations with high proportions of optimal individuals. However, the discovery of attracting fixed points in the neighborhood of optimal populations is not always sufficient, especially in the case of dynamically changing fitnesses. As an example, in Section 6.7 we study an evolutionary model that exhibits cyclic limiting behavior. We introduce specific methods for specific models at the appropriate places, where these specific methods are required. These and other methods are discussed in Vose (1999b).

Finite populations. A Markov model $\langle \pi, \tau \rangle$ with transition matrix T has a unique limit or steady state distribution over its states if the transition matrix is irreducible and aperiodic (i.e., ergodic). In the case of ergodicity, the stochastic transition probability matrix has a unique stochastic eigenvector $\mathbf{x}^*$, with eigenvalue 1 with

$$\mathbf{x}^* = \lim_{g \to \infty} T^g \mathbf{x}(0). \tag{3.32}$$

This limit distribution is independent of the initial distribution $\mathbf{x}(0)$ if transition matrix T is irreducible and aperiodic. This is stated by the Perron-Frobenius theorem, which is introduced and discussed in detail in Appendix B.

If we want to determine the limit behavior of an SGA, the following theorem shows that the assumption of a mutation rate μ with $0 < \mu < 1$ is a sufficient prerequisite for the Markov model of the SGA to be irreducible and aperiodic.

Theorem 3.2. *If $0 < \mu < 1$ holds for mutation rate μ of an SGA, then transition matrix T of the corresponding SGA is irreducible and aperiodic.*

Proof. If $0 < \mu < 1$, the probability, in (3.8), that any string in Ω_S is mutated to any other string in Ω_S is always nonzero. As a consequence, the probability that any two strings in Ω_S are recombined to form any daughter string in Ω_S through meiotic operator θ in (3.14) is also nonzero. For both the haploid and diploid SGA, the probability to generate any new child individual in Ω, from (3.15) and (3.16), is also nonzero since at least one genome in the model has a strict positive selection probability. As a result, all factors in (3.20) are nonzero. Consequently, all probabilities, to transit from any population $P \in \pi$ to any other population $P' \in \pi$ is strictly positive, such that the transition probability matrix contains entries that are all nonzero.

A probability matrix for which all transition probabilities are strictly positive, is irreducible and aperiodic, thus concluding the proof of this theorem. □

As a corollary to this theorem, the unique limit or steady state distribution $\mathbf{x}^*$ of the SGA exists if the mutation probability μ is strict positive and not 1. In practice, mutation rates are commonly rather small and we generally assume that the mutation rate for bit strings is within the range $0 < \mu < 0.5$. For mutation rates $\mu = 0.5$ and larger, the mutations in the evolutionary process work against the progress made by selection in the model, since it causes the majority of fit alleles to be mutated.

In the case that $\mu = 0$, the system becomes reducible. Indeed, if a genome is lost from the population, it can no longer return to the population as a mutant, which in its turn means that some states of the system can no longer be reached. If some states are unreachable from other states, the model is reducible and the limit behavior is then dependent on the initial distribution over the population. In the case that $\mu = 1$, the system becomes periodic. Indeed, if a population consists of individuals of the same genome, all alleles present mutate, generation after generation, resulting in a periodic behavior for which no limiting behavior exists.

3.8 Numerical Methods

3.8.1 The power method

We now face the problem of numerically determining the eigenvector with corresponding eigenvalue 1 of the transition probability matrix T. There are many algorithms for approximating the eigenvectors and eigenvalues of $N \times N$ matrices, see for an overview e.g., (Wilkinson 1965; Parlett 1980; Saad 1992). Algorithms that are based on factorizations compute the complete eigenpair spectrum of a matrix. Since the factorization methods cost $O(N^3)$ operations, and as we are only interested in the eigenvector of the matrix with dominating eigenvalue instead of the other unnecessary eigenpairs, they are not convenient for our goal. We use the power method, based on matrix-vector multiplications since these operations are cheap, and the power method renders eigenvalue-eigenvector pairs instead of complete spectra (Kuczyńsky & Woźniakowski 1992; Del Corso 1997).

Because we know beforehand that the dominating eigenvalue of the transition probability matrix equals 1 (in case of ergodicity), a simple version of the power method can be implemented, as the eigenvalue is not to be computed. The power method for an ergodic transition probability matrix T is given by the following algorithm.

Algorithm 3.2 (Power method for stochastic and ergodic matrices).

```
$g := 0$
choose $\mathbf{x}(0)$ as the initial guess
**repeat**
  $g := g + 1$
  $\mathbf{x}(g) := T\mathbf{x}(g-1)$
**until** $d(\mathbf{x}(g), \mathbf{x}(g-1)) < \epsilon$
accept $\mathbf{x}^* = \mathbf{x}(g)$ as the eigenvector with dominating eigenvalue 1
```

In this algorithm, $d(\mathbf{x}, \mathbf{y})$ denotes a suitable error measure between two vectors, such that a termination criterion can be implemented in the power method to recognize numerical convergence of the system. The algorithm is stopped upon detection of changes smaller than ϵ. Some examples for the power method are the following, as discussed by van Heeswijk (2004).

- Maximum absolute element-wise change, with

$$d_1(\mathbf{x}, \mathbf{y}) = \max_{i \in \pi} |x_i - y_i|$$

- Maximum relative element-wise change, with

$$d_2(\mathbf{x}, \mathbf{y}) = \max_{i \in \pi} \frac{|x_i - y_i|}{x_i}$$

- Maximum relative change with respect to the maximal element, with

$$d_3(\mathbf{x}, \mathbf{y}) = \max_{i \in \pi} \frac{|x_i - y_i|}{x_{max}} \text{ with } x_{max} = \max_{i \in \pi} x_i$$

- Sum of absolute element-wise change, with

$$d_4(\mathbf{x}, \mathbf{y}) = \sum_{i \in \pi} |x_i - y_i|$$

The eigenvector $\mathbf{x}^*$ of the transition probability matrix can thus be approximated numerically by implementing the iterative process of the power method. An initial distribution $\mathbf{x}(0)$ is chosen, and the consecutive distributions $\mathbf{x}(g)$ are computed by repeatedly multiplying transition probability transition matrix T

with this vector. Because of the Perron-Frobenius theorem, the consecutive distributions converge to the limit distribution $\mathbf{x}^*$. In a numerical approximation, we keep track of the changes between consecutive distributions, and a stop criterion is implemented that recognizes the convergence of the numerical iteration.

In the general power method, two problems may arise. If the starting vector has a zero component in the direction of the eigenspace corresponding to the largest eigenvalue, there is no convergence to the eigenvector with largest eigenvalue, but to the eigenvector with second largest eigenvalue. However, in the case of ergodicity in a transition probability matrix, and if the initial vector is stochastic, this problem does not arise as a corollary of the Perron-Frobenius theorem, see Appendix B. In more general cases, random techniques with multiple start vectors can be applied to correctly estimate the largest eigenvalue and its corresponding eigenvector, as first proposed by Shub (1986) to improve efficiency, used by Kostlan (1988) to study performance, and extensively studied by Del Corso (1996, 1997). In order to increase the efficiency of the power method for finding the eigenvectors with dominating eigenvalues of transition probability matrices, similar techniques can be applied.

As a second problem, the speed of convergence depends dramatically on the second largest eigenvalue of the transition probability matrix. The rate of convergence for estimating the dominant eigenvector and its eigenvalue is $|\lambda_2/\lambda_1|^g$ for the gth iteration, with λ_1 and λ_2 being the largest and second largest eigenvalues. As a result, slow convergence is expected when the separation of the eigenvalues is badly bounded. Note that the convergence of the Markov chain is a reflection of the rate of convergence of an instantiation of the GA represented by the model, see e.g., Vitányi (2000).

3.8.2 Transition matrix size

When implementing the iterative process to determine the eigenvector of the transition matrix T, the transition matrix T is computed beforehand, and kept in memory during the iterative process. Since the number of populations grows quickly with increasing size of search space Ω and population size r, see also (3.4), the size of the matrix also becomes easily too large to be kept in memory.

Table 3.1 gives the number of possible population configurations, or states of the Markov chain, for some (small) values of the search space and population size, as given by (3.3).

If the transition probability matrix is implemented with the `double` data type, which commonly consists of 8 bytes, then the Table 3.2 gives the required memory size in bytes b, with $b = 8 * |\pi|^2$ to contain transition matrix T in memory.

As an example from this table, a haploid SGA with small population size $r = 10$ and a search space of bit strings with length $l = 3$, i.e., with search

$\|\pi\|$	$\|\Omega\| = 2$	$\|\Omega\| = 4$	$\|\Omega\| = 6$	$\|\Omega\| = 8$
$r = 10$	11	286	$3,003$	$19,448$
$r = 20$	21	$1,771$	$53,130$	$888,030$
$r = 40$	41	$12,341$	$1,221,759$	$62,891,499$

Table 3.1: State space size of the SGA

b	$\|\Omega\| = 2$	$\|\Omega\| = 4$	$\|\Omega\| = 6$	$\|\Omega\| = 8$
$r = 10$	968 $\times 2^{0}$	639.03 $\times 2^{10}$	68.80 $\times 2^{20}$	2.82 $\times 2^{30}$
$r = 20$	3.45 $\times 2^{10}$	23.93 $\times 2^{20}$	21.03 $\times 2^{30}$	5.74 $\times 2^{40}$
$r = 40$	13.13 $\times 2^{10}$	1.13 $\times 2^{30}$	10.86 $\times 2^{40}$	28.10 $\times 2^{50}$

Table 3.2: Transition matrix of the SGA, size in bytes

space size $|\Omega| = 8$, already requires 2.82GiB* of memory to contain the transition probabilities of the SGA. As a result, the size of the transition probability matrices can easily get out of hand (for computer memories that are currently common to reside around GiBs), even for small population sizes and small search spaces.

3.8.3 Transition matrix compression

In order to decrease the size of the matrices, various algorithms have been proposed to lump states of the model together, thus reducing the overall size of the transition matrix. As the size of the matrices is reduced, computing the limit behavior becomes a more feasible computational task. The resulting behavior (e.g., the limit behavior of the compressed system) is then unpacked to render an approximation of the behavior of the system as a whole. Indeed, as the matrix is compressed by grouping states together, the compression algorithm may introduce an error such that exact computations of the systems' behavior is no longer possible. A short overview of existing compression techniques is given.

Exploiting symmetries in the transition matrix. In specific cases, symmetries in the probability transition matrix can be found. Because of these symmetries, duplicate information is stored in the transition matrix. As an example, if two genotypes in the search space of the genetic algorithm have the same fitness values, and their probabilities of being reproduced to become other genotypes are the same, then populations, or states with N individuals of the first genotype

*According to S.I. standards on prefixes for binary multiples, we use KiB (kibibyte) to represent 1024 bytes, MiB (mebibyte) to represent 1024^2 bytes, GiB (gibibyte) to represent 1024^3 bytes and TiB (tebibyte) to represent 1024^4 bytes, instead of the common, but confusing kilo, mega, giga and tera prefixes (Barrow 1997; IEC 2000), which represent powers of 1000, not of 1024 have the same behavior as the corresponding populations with N individuals of the second genotype instead of the first genotype. In such specific cases, the transition probability matrix contains redundant column vectors which are permutations of each other. By discovering and carefully organizing the structures of these redundancies, and accounting for their permutations, a transition matrix, and its computations, can be reduced in a relatively complicated implementation. Since such a compression technique is very specific with respect to the population structures, the selection process, and reproduction in the underlying model, the necessary work on the implementation may not be in relation to the gains of compressing the specific transition probability matrix. This is especially the case if a multitude of models has to be studied. Note that this compression technique is lossless and does not introduce an error in the resulting behavior. A more general, lossy, method of compressing the transition probability matrix is in many cases an easier and more productive option. Rowe & Vose & Wright (2005) present a general framework for compression under which aggregation can take place with no loss of information, and shortly discuss the applications of this framework on the aggregation of states in a Markov model of a basic genetic algorithm.

Aggregating states with similar behavior. An initial lossy compression algorithm for models of genetic algorithms was presented by Spears et al. (1996, 1998a, 1998b, 1999). Their algorithm aggregates states with similar behavior to decrease the size of the transition matrices of Markov models of genetic algorithms. Spears' algorithm starts out with the complete matrix. Repeatedly, populations from the state space are aggregated if their distributions at the next generation are similar, i.e., if their corresponding column vectors in the probability transition matrix are close together. This process is repeated until a feasible matrix, i.e., with a feasible size, has been generated.

A problem with this technique is the fact that the exact transition matrix as a whole must be computed and placed in memory before the compression algorithm can start its work. Since the size of the matrix, and the required memory to store the matrix, is the reason why compression algorithms are designed in the first place, this algorithm becomes purposeless if huge matrices need to be studied. Note that instead of storing the original matrix in memory, the implementation of this aggregation algorithm can also re-generate the columns as they are required for finding suitable states to aggregate, but this would require an immense amount of recomputation of the column vectors in order to find a reasonably small compressed version of the original exact transition matrix. Indeed, if there are $N \gg 0$ states in the uncompressed model, $O(N^2)$ columns (each of size N) need to be computed to find the two states that are most similar, requiring $O(N^2M)$ recomputations of columns if the state space has to be reduced by M states.

Aggregating similar states. An alternative compression technique is introduced by Moey & Rowe (2004a, 2004b). This compression technique looks at properties of the populations and decides on similarities between these properties which states of the state space can be aggregated. As such, the aggregation of states can be decided upon without generating the original exact probability transition matrix, undoing Spears' problem of having to store or recompute the matrix at every aggregation step. Moey and Rowe adopt similarities in properties of fitness of the populations (maximal fitness and average fitness), or the fact that states are near a similar fixed point of heuristic $\mathcal{G}$ to cluster similar states together. Assuming the hypothesis that finite population genetic algorithms spend most of their time near the fixed points of $\mathcal{G}$ (Vose 1999b), allows for a fairly good compression of the matrices, while introducing relatively small errors.

Small mutation rates introduce large errors. A general, more troublesome problem with lossy techniques is of concern for the goal of this thesis. Both Spears & De Jong (1996) and Moey & Rowe (2004a) point out that lossy aggregation algorithms introduce larger errors for smaller mutation rates. As the focus of this thesis, genetic drift, forces us to mainly study models with very small mutation rates, we are consequently forced not to rely on these aggregation algorithms, and we are required to work with the exact, large transition probability matrices.

3.8.4 Parallel implementation of the power method

Since probability transition matrices can easily grow too large to be analyzed by a single computer (according to the current common standards of single computers having a memory of *merely* 1 or 2 gibibytes), and as we don't want to be restricted easily by this limitation, we need a way to parallelize the analysis such that the power method can be ran as a parallel method.

Our research group for BioModeling and BioInformatics has access to a Linux cluster, named *Biowulf.* This cluster has, besides a master node, 36 slave nodes. Each of these nodes accommodates two `AMD 1800+` processors, `2GiBs` of memory, and two network cards to connect to a main communication bus.

The power method can easily be partitioned to run as a parallel computation. This section gives a general description of our method of parallelization.

Parallelization of the power method. During his internship at our group, van Heeswijk (2004) has used the *Biowulf* cluster to implement a parallel version of the power method for studying the transient and limit behavior of Markov models of genetic algorithms. The power method repeatedly multiplies transition matrix T with the current vector denoting the distribution $\mathbf{x}(g-1)$ to become the distribution $\mathbf{x}(g)$ at time step g. Matrix-vector multiplications can easily be partitioned row- or column-wise, to accommodate a parallel implementation.

Row-wise partitioning. Let $\mathbf{r}_i^\top$ denote the ith row in a $N \times N$ matrix T. We can write the matrix-vector multiplication $T\mathbf{x}$ row-wise with

$$T\mathbf{x} = \begin{pmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \\ \vdots \\ \mathbf{r}_{N-1}^\top \\ \mathbf{r}_N^\top \end{pmatrix} \mathbf{x} = \begin{pmatrix} \mathbf{r}_1^\top\mathbf{x} \\ \mathbf{r}_2^\top\mathbf{x} \\ \vdots \\ \mathbf{r}_{N-1}^\top\mathbf{x} \\ \mathbf{r}_N^\top\mathbf{x} \end{pmatrix}. \tag{3.33}$$

As a result, we can partition T into blocks of rows, and store each block in the memory of a process of the parallel method. For the repetitive steps in the power method, we let each of the processes compute their corresponding block of elements of $T\mathbf{x}$. The blocks of the new vector are then merged, and redistributed among the nodes to compute the new iteration step of the power method. At each time step, all of the processes in the parallel program receive the new vector $\mathbf{x}$ and can individually decide whether the power method has converged.

Column-wise partitioning. Let $\mathbf{c}_i$ denote the ith column in a $N \times N$ matrix T. We can write the matrix-vector multiplication $T\mathbf{x}$ column-wise with

$$T\mathbf{x} = \begin{pmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_{N-1} & \mathbf{c}_N \end{pmatrix} \mathbf{x} = \sum_{i=1}^{N} x_i \mathbf{c}_i \tag{3.34}$$

As a result, we can partition T into blocks of columns, and store each of these blocks into the memory of a process in the parallel program. For the repetitive steps in the power method, we let each of the processes compute its contribution of $T\mathbf{x}$. These contributions are then merged, and redistributed among the nodes to compute the next iteration step of the power method. Similar to the row-wise partitioning, each of the processes is able to decide whether convergence has been reached or not.

In the parallel implementation of van Heeswijk (2004), column-wise partitioning is chosen, for two reasons. In the first place, the computation of a column vector of the probability transition matrix can be done quickly within one process. Indeed, a column vector of the matrix is a multinomial distribution according to (3.20). Consequently, each of the processes in a column-partitioned parallel program can be independently responsible for the computation of its column vectors. In the row-wise partitioning method each of the processes would need to compute or communicate $\mathcal{G}$ for all of the possible finite populations, which is slower than restricting the computation of $\mathcal{G}$ to only those states that are required for generating the required column vectors. Secondly, van Heeswijk uses LZO compression (Oberhumer 2002) to further (lossless) compress the binary data describing a column to be able to store more data in memory. The compression method is slow, but the unpacking of the data of column vectors, which is required at each iteration of the power method, is very fast. The compression of columns of the transition matrix has been experimentally observed to be better than that of rows.

Results. As a result of this parallelization of the power method, fairly large transition matrices (according to current standards) can be analyzed. In his report, van Heeswijk reports results of genetic algorithms with population size $r = 13$ over a search space of $|\Omega| = 8$ genomes. These experiments correspond to probability transition matrices of 44.77`GiB`, being compressed with LZO to 24.7`GiB`. The biggest experiments ran used a transition probability matrix of 118`GiB`, for analyzing the limit behavior of a population of $r = 500$ individuals over $|\Omega| = 3$ individuals. This is far beyond any current standings in the lossless study of Markov chains of GAs. The computation of this last experiment required 2 full days on the *Biowulf* cluster to find the converged limit behavior. In the parallel method, a speedup of about 90% was obtained, i.e., as an example, the power method runs 18 times faster on 20 processors as compared to running it on a single processor. More detailed information on these and similar results can be found in (van Heeswijk 2004).

As a result, a parallel implementation of the power method allows us to exactly study fairly large systems (in number of states), without loss of accuracy due to lossy compression and approximation.

Chapter 4

Static Fitness Environments

We provide expectations of long run behavior of finite and infinite populations in the evolutionary models we have constructed in the previous chapter, adopting an experimental mathematics approach. We focus on expectations of the limit behavior of systems with individuals with one locus and two alleles whose genotypes have a fixed fitness. These basic models provide a general trend for the expectations of long run behavior for models with varying selective pressure, population size and mutation rate. Most of the results supplied in this chapter are not new, per se, but provide a basis for understanding the observations made in the applications of dynamically changing environments in the following chapters.

We first study a finite population model with no selection, which is similar to the model in the original study on genetic drift by Wright (1931). For both haploid and diploid populations, the influence of mutation rate and population size is discussed. We provide a correction of one of Wright's initial results on the balance between drift and mutation. We then introduce selection into the model, and review the influences of selective pressure, population size, mutation rate and ploidy on the long run behavior of finite populations.

4.1 Introduction

Before studying models of finite populations with fitness determining environments that change over time, we give some results of environments where the fitness of an individual remains static over time. Models similar to the ones in this chapter are later combined to create dynamically changing fitness functions. An overview of the influence of parameters on the long run behavior of static fitness functions is thus in place before moving to models of dynamically changing fitness environments.

We start off with the simplest possible model of individuals with one locus and two alleles, where no selection or variation is present, and later drop these assumptions, by incorporating variation and selection. This allows us to give an overview of the influence of the different parameters in the models on genetic drift, and their relation to random sampling.

Similar models are studied in the areas of population genetics and the theory of genetic algorithms. For overviews of these models and specific applications that are out of scope for this text, we would like to refer to the population genetics overview given by Ewens (2004), and an overview of the transient behavior of finite population Markov models in applications of genetic algorithms in function optimization, e.g. De Jong et al (1995).

We have to point out that we mainly study the long term behavior of the systems, and generally do not consider the transient behavior of the systems. Also, we ignore the time required for these systems to reach their steady state distributions. We do not discuss convergence time in detail, but nevertheless recognize the importance thereof for the study of evolutionary systems. We refer the reader to models that study the time to fixation in models with no selection, such as the models with diffusion of Kimura & Otha (1969), or the application of rapidly mixing of the transition probability matrices, required for polynomial time convergence to populations with a high degree of highly fit individuals in Vitányi (2000).

4.2 One Locus, Two Alleles, No Selection

4.2.1 Model

In this first model, we consider haploid populations of individuals with one locus, and two alleles. An individual can thus be represented with 1 bit, i.e., $\Omega = \{0, 1\}$. We also assume that there is no selective pressure among different individuals, i.e., the fitness of all individuals $i \in \Omega$ is the same constant c, with $f(i, P) = c$. We adopt this model for control measurements of later applications, and to show how a population behaves in the absence of selection. The predictions of this first model give us an idea of how strong genetic drift can become for certain parameters, as variation with mutation rate μ and sampling of the population are the only processes at work in systems with neutral selection.

Note that predictions of this model correspond to the behavior of a diploid population, with one locus, two alleles and no selection. Indeed, we have pointed out in Section 2.3.2 that the only difference between the reproduction scheme of haploid and diploid populations is the state in which individuals are mature and when fitness is determined. Since selective pressure is assumed not to be present in this model, the dynamics of a diploid population of size r thus corresponds with a haploid population of size $2r$, where the mature individuals in the haploid populations coincide with gametes in the diploid population. The predictions of this section consequently also relate to diploid models, whose population size is half that of the haploid models.

Reduced heuristic function. Let $\mathbf{p} = (p_0, p_1)^\top$ represent an infinitely large haploid population P for this model, which is only subject to mutation. Since the genotypes of the individuals only consist of one locus with only two possible, atomic alleles, crossover can be ignored. Accordingly, we can reduce the heuristic function $\mathcal{G}$, as constructed in the previous chapter considerably, with

$$\mathcal{G}(\mathbf{p}) \;=\; \left(\quad (1-\mu)p_0 + \mu p_1 \quad , \quad \mu p_0 + (1-\mu)p_1 \quad \right)^\top$$

$$\begin{aligned} &= (\ p_0(1-\mu)+(1-p_0)\mu\ ,\ (1-p_1)\mu+p_1(1-\mu)\ )^\top \\ &= (\ (1-2\mu)p_0+\mu\ ,\ (1-2\mu)p_1+\mu\ )^\top. \end{aligned} \tag{4.1}$$

Since $p_1 = 1-p_0$ for any population P, we can thus write a step of this system with an infinite population size only in proportions of genome 0, with $p_0' = (1-2\mu)p_0+\mu$ representing the expected proportion of genotype 0 in population P' at the next generation for a given population P.

4.2.2 No variation

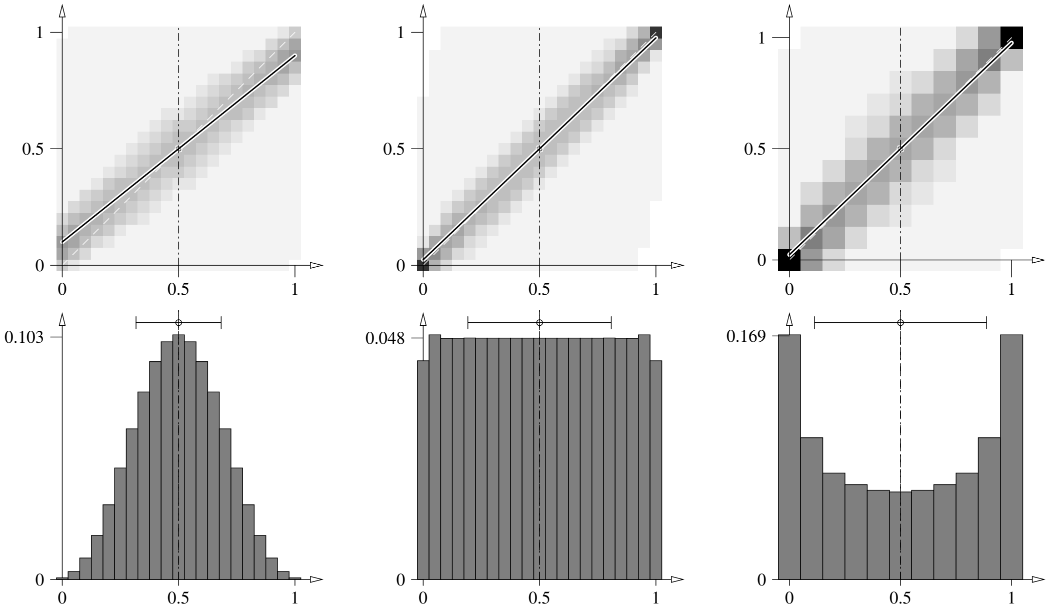
We first assume that no variation is present, i.e., $\mu = 0$.

In this case, all states of the infinite model are fixed points of the system. Indeed, in this case (4.1) can further be reduced to $\mathcal{G}(\mathbf{p}) = \mathbf{p}$, indicating that no change in proportions of types 0 and 1 is expected over time, and that consequently all states are fixed points. None of the fixed points are stable, nor unstable in the usual sense; a small perturbation from one fixed point does not die out, it simply gives a different solution.

If the assumption of infinitely large populations is dropped, and a finite population size is assumed, then the populations with $p_0 = 0$ or $p_0 = 1$ – the populations that only contain one allele, either all 1 or all 0, respectively – are steady states. The probability transition matrix T of this system, as given by (3.21), is reducible. Indeed, if the system ends up in either of the two states where $p_0 = 1$, or $p_1 = 1$, then no escape from these states is possible, making the chain reducible. As a result, there is no unique eigenvector with eigenvalue 1 of transition probability matrix, that would give the unique limit distribution of the system at hand. Depending on the initial distribution $p_0(0)$ and $p_1(0)$ of alleles 0 and 1, the system has probability $p_0(0)$ of ending up in population P^* with all individuals of type 0, i.e., $p_0^* = 1$, in the long run, and probability $p_1(0)$ to end up in the state with $p_0^* = 0$, see, e.g., Wright (1931).

Note that the differences in predictions between the infinite population model and the finite population model are due to genetic drift, by definition. Where the deterministic infinite model predicts that the initial distribution among individuals is expected to be stable, the stochastic, finite model predicts that the populations drift away from these proportions, to become fixated in a population that consists either of all 0 or all 1 individuals.

These observations relate to the Hardy-Weinberg equilibrium. If no variation or selection is present in the model, the frequencies of genotypes in the infinitely large model remain in their initial frequency ratio $p_0 : p_1$. For the corresponding infinitely sized diploid model, diploid individuals with genotypes $\{0,0\}$, $\{0,1\}$ and $\{1,1\}$ are created through fertilization with ratio $p_0^2 : 2p_0p_1 : p_1^2$, and this ratio remains the same over time, which is stated by the Hardy-Weinberg principle (Hardy 1908; Weinberg 1908). In contrast, the ratios of frequencies of the diploid genotypes in finitely sized population drift away in similarity with the haploid



(a) $r = 20, \mu = 0.1$ (b) $r = 20, \mu = 0.024$ (c) $r = 10, \mu = 0.024$

Figure 4.1: Graphic representations of transition functions (top) and steady state distributions (bottom) for the system with no selective pressure for 3 different parameter settings. See the text for the key of these figures.

model, until their ratio hits either one of the boundaries at $1:0:0$ or $0:0:1$. The probability of ending up in the state that is identified by ratio $1:0:0$ is equal to the initial frequency $p_0(0)$ of gamete 0, see, i.e., Ewens (2004).

4.2.3 Variation

If a variation operator – which is symmetric for both strategies – with $0 < \mu < 1/2$ is assumed, only the state $\mathbf{p}^* = (1/2, 1/2)$ is a fixed point of the infinite population model, given by the solution of

$$\begin{aligned} p_0^* &= (\mathcal{G}(\mathbf{p}^*))_0 \\ &= (1-2\mu)p_0^* + \mu \end{aligned}$$

which holds for $p_0^* = 1/2$. This fixed point is also stable, since the derivative of $\mathcal{G}$ in $\mathbf{p}^*$ is smaller than 1, making the fixed point an attractor of the system. An infinite model thus predicts that the system ends up in a population that has an equal amount of individuals with phenotype 0 and those of type 1. We now contrast this prediction with expectations of finite, stochastic population models.

In Theorem 3.2, we have shown that for $0 < \mu < 1$, the finite population model is ergodic. Consequently its limit behavior is given by the unique stochastic

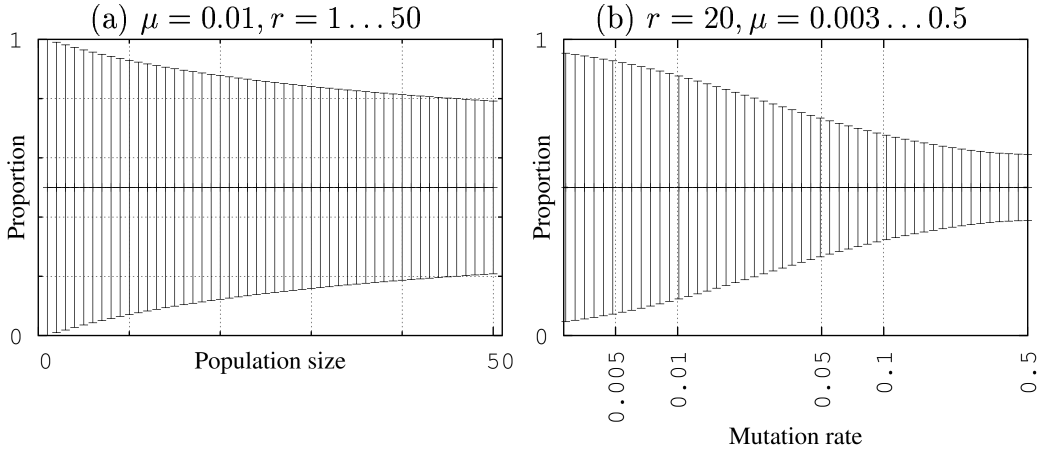


Figure 4.2: Expected proportion of 0 genomes and standard deviation about this expectation in the limit distribution of the system with no selective pressure, for (a) different population sizes with $\mu = 0.01$ and (b) different mutation rates with $r = 20$.

eigenvector of the transition probability matrix of the system, with corresponding eigenvalue 1. This distribution over the states of the system describes the long run behavior. For the system with finite populations, Figure 4.1 depicts the transition functions and the limit or fixed point distributions of the finite population model, for a small set of population sizes and mutation rates.

The top figures represent the transition matrices of the finite population model. The deterministic map diagram of the infinite population model defined by $\mathcal{G}$ is also shown. The horizontal axis represents the current proportion of the genomes of type 0 in the population, the vertical axis represents the proportion at the next generation. Each gray scaled box represents the transition probability between states in one generation for the finite model. Darker grays represent higher probabilities. The gray scale is the same for all three figures. The infinite model's mapping function ($\mathcal{G}$) is imposed on each of the figures. The dashed white line represents the diagonal, i.e. where the proportion of 0 genomes at the next generation equals the proportion at the current generation.

In the bottom figures, the bars denote the probability of ending up in the population with the given proportion of 0 genomes. The dashed vertical line denotes the mean of the distribution, i.e., the expected proportion of individuals of type 0 according to the distribution. This expected mean is always at 0.5 because of symmetries in the system, and is thus equal to the fixed point of the infinite population model. The horizontal error bars denote the expected deviation of the finite model's limit behavior with respect to the infinite population's fixed point (see Section 3.7.2). Since the fixed point of the infinite model is equal to the mean of the distribution, it also represents the standard deviation of the distribution according to the mean of the distribution. Note that there is no standard deviation in the infinite model as the system is deterministic and has a unique fixed point.

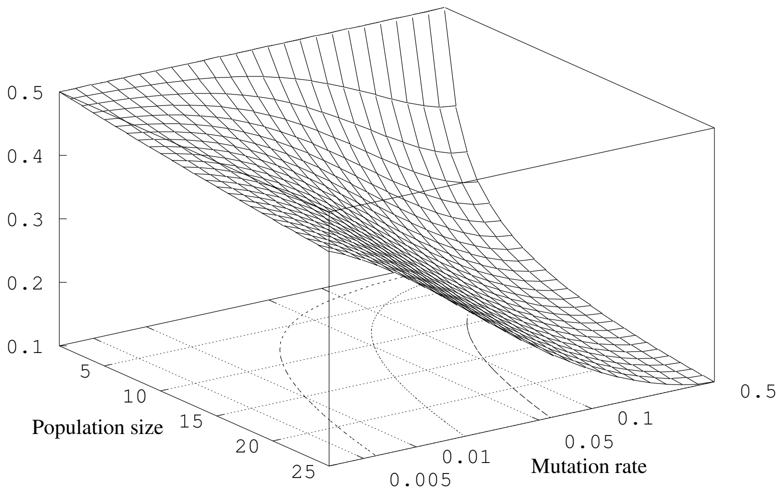


Figure 4.3: Standard deviation about the mean of the proportion of 0 genome in the limit distribution of the system without selective pressure, for different population sizes and mutation rates. Expected proportions of the 0 genome is always 1/2.

Figure 4.1(a) shows a typical steady state distribution for sufficiently large population sizes and large mutation rates. In this type, the system is most likely to end up with highly diverse populations.

Figure 4.1(c) shows a typical limit distribution for systems with a sufficiently small population size or a relatively small mutation rate. In these cases, a run of the system will most likely end up in either one of the populations filled exclusively with either genome 0 or 1. Note that with these parameter settings, the system prefers extremes of the state space, and avoids the predicted “stable” fixed point of the infinite population model. Similar behavior has also been observed in models of simple genetic algorithms (Vitányi 2000). An evolution, or instantiation of the model diverges from the expected average behavior of the infinite model, and even from the finite model’s expected behavior. Once a population becomes fixated in either all 0s or all 1s, chances are small that the population escapes from this state of fixation to return to the expected average behavior. It is often assumed, when studying models of infinite populations that perturbations from the infinite model – as a result of stochastic sampling – fade out after time and return to the state as predicted by the infinite model. However, since stochastic perturbations develop continuously in a small population, there is only a very small chance for the population to return to its expected behavior. As a result, the variance of the expected average behavior is larger.

Figure 4.1(b) shows a snapshot of the transition from the first type to the second. Note that the behavior depicted in these distributions is structurally very different, although the infinite population model predicts the same stable fixed point for all of these evolutionary systems. These differences in predicted behavior are due to genetic drift around this stable fixed point. Drift is stronger as populations become smaller, since more sampling effects govern the dynamics of the system. Drift is also stronger as the mutation rate decreases, which results in a higher probability of fixation of an allele due to slow production of genetic diversity.

The standard deviation σ of the steady state distributions' means can be employed to discuss the importance of genetic drift with finite populations, and we are able to predict how the parameters are influencing the system's behavior. Later on, we use similar techniques to discuss genetic drift in evolutionary systems with selective pressure, and can use the current model – which assumes no selective pressure – as a control measurement for predictions and expectations in the presence of selection.

Two forces are at work in the system, and try to keep each other in balance. On one hand, mutation introduces diversity in the populations, and pushes the population to a ratio of frequencies 0.5 : 0.5, as in the infinite model. On the other hand, random sampling in the finite model causes the frequencies of the genotypes to drift away from the expected equilibrium of the infinite population model, at ratio 0.5 : 0.5, toward populations in which either allele 0 or 1 is fixated. Dependent on whether random sampling or variance through mutations is strongest, higher probabilities are expected at the edges of the limit distribution – representing a high probability of ending up in populations which fixated on a specific allele – or near the middle of the limit distribution – representing a high probability of ending up in a population that is highly diverse – respectively. We give an overview of the interaction of varying population sizes and mutation rates.

Population size. Figure 4.2(a) shows the influence of a varying population size on the standard deviation of the distribution around the mean, for $\mu = 0.01$. Figure 4.3 depicts the influence of the population size for a range of mutation rates.

Assume a fixed mutation rate μ with $0 < \mu < 1/2$. As population size r increases, the standard deviation σ decreases. Indeed, as populations become larger, the effects of random sampling, and thus genetic drift in the finite population, is less apparent, and the system as a whole becomes more deterministic. Consequently, as the population size becomes larger, the finite population model behaves more similar to the deterministic infinite population model, where $\sigma = 0$. Indeed, if we keep on increasing the population size, then σ comes closer to 0, and the mean of the distribution converges to the fixed point of the infinite population model. Note that in the current model, the fixed point of the infinite model and the mean of the finite population are always the same. This statement is thus trivial, but becomes more important as there will be a difference between these two predictions as selective pressure is considered.

In the case that the population size goes toward the other extreme, where $r = 1$, the system becomes more influenced by genetic drift. As $r = 1$, the only two states of the system are states of fixation, and either state is visited with probability 1/2 in the long run distribution. As the population size goes toward this extreme, so does the corresponding system's behavior, independent of the mutation rate.

Mutation rate. Figure 4.2(b) shows the influence of a varying mutation rate on the standard deviation of the distribution around the mean, for $r = 20$. Figure 4.3 depicts the influence of the mutation rate for a range of population sizes.

Assume a fixed and finite population size r. As the mutation rate in the evolutionary system is sufficiently decreased, a run of the system most probably ends up in either one of the populations with only one genome, as the rate of new mutations being added to the population doesn't keep up with the rate of fixation due to random sampling. In the limit, where a mutation rate of 0 is assumed, the transition matrix of the evolutionary system becomes reducible, and the steady state distribution of the Markov chain is no longer unique. If $\mu = 0$, two linearly independent stochastic eigenvectors of the transition probability matrix with corresponding eigenvalue 1 exist, and either one of these eigenvectors represents a distribution where the evolutionary system ends up with a population containing only one genotype. As the mutation rate is sufficiently decreased toward 0, the finite population model better resembles the extreme situation where $\mu = 0$.

As the variational pressure is increased, i.e., as μ becomes larger, more random individuals are generated by the reproduction process. In the extreme case, where $\mu = 1/2$, each generation renders a new random population with the probability of either individual in this population being 1/2. At each step, a distribution over the state space is constructed that is a binomial distribution. The probability of encountering a population with n out of r individuals being of type 0 is then given by $\left(\begin{array}{c} r \\ n \end{array} \right) \frac{1}{2}^r$. As this is the case for each of the generations during a run of the system, it is also the limit or steady state distribution. Consequently, if a mutation rate of 1/2 is assumed, the expected proportion of either genome is 1/2. The standard deviation of a binomial distribution of r trials over probabilities p and $q = p - 1$ about its mean equals $\sqrt{rpq}$. When scaled down to proportions, the standard deviation σ about our expectation thus becomes $\frac{1}{2\sqrt{r}}$. As the mutation rate is increased toward 1/2, the finite population model better resembles the extreme situation where $\mu = 1/2$.

With higher mutation rates, the transition probability to go from any particular population to any other is more evenly distributed over the set of states. As more variational pressure is available in the model, the time required for the power method to hit a good approximation of the limit behavior becomes shorter.

Relation between population size and mutation rate. According to the observations of the previous sections, there seems to be a strong connection between the mutation rate and population size of the model. For a given population size, a sufficiently low mutation rate can be found such that the expected behavior is governed by fixation of alleles in the population. Similarly, a relatively large mutation rate results in highly diverse populations. At a specific mutation rate – which is a function of the population size – the behavior transits from the first behavior pattern to the second. As the population size is increased, the mutation rate at which this transition takes place is decreased, since the effects of random sampling are less strong in large populations and can thus more easily be countered by a low mutation rate.

In the continuous approximation of the model, as studied by Wright (1931) and given full account by Ewens (2004), a general formula that gives a description of the limit distribution $\mathbf{x}^*$ is given by

$$(\mathbf{x}^*)_P = Cp_0^{2r\mu-1}(1-p_0)^{2r\mu-1}, \tag{4.2}$$

with C a normalization constant given by

$$\frac{1}{C} = \sum_{i=0}^{r} \left(\frac{i}{r}\right)^{2r\mu-1} \left(1-\frac{i}{r}\right)^{2r\mu-1}. \tag{4.3}$$

Note that this form of the limit distribution depends on the product of population size r and mutation rate μ.

For large values of $r\mu$, the approximation of the limit distribution as given by (4.2) has a maximum at $p_0 = 1/2$, and minimal values at both $p_0 = 0$ and $p_0 = 1$. This corresponds to a limit behavior that has high probabilities of populations that are highly diverse, and a low probability of fixation of the population.

For small values of $r\mu$, the limit distribution in (4.2) has a minimum at $p_0 = 1/2$, and maximal values at both $p_0 = 0$ and $p_0 = 1$. In such a distribution, there is a high probability of populations that are fixated, and a low probability of being in a state whose population is highly diverse.

If μ and r are chosen according to $\mu = 1/2r$, (4.2) predicts a uniform limit distribution.

From this approximation, Wright derives that if $\mu < \frac{1}{2r}$, then genetic drift is stronger than the rate of introducing new mutants in the population, resulting in a distribution that has its maximal probabilities for the populations with either $p_0 = 1$ and $p_1 = 1$. If $\mu > \frac{1}{2r}$, the rate of mutation is stronger than the rate of fixation, and the distribution is predicted to have a maximum probability at $p_0 = 1/2$.

Wright and Ewens thus predict that the transition between these two possible situations occurs at $\mu = 1/2r$.

Although Wright predicts uniform equilibrium distributions at this point, numerical computations show that the exact equilibrium distribution for $\mu = \frac{1}{2r}$ is not really uniform in this case, see also Figure 4.4(a), which we discuss in more detail later on. Hence, we investigate whether a uniform distribution for specific values of r and μ is possible. In fact, an equilibrium distribution $\mathbf{x}^*$ with all components x_i equal, i.e., $x_i = \frac{1}{r+1}$ cannot be found. However, a limit distribution with all components x_i approximately equal, except for i very small and very large, does exist if $\mu = \frac{1}{2r+2}$. We give the derivation of this result.

The equilibrium distribution, or limit distribution $\mathbf{x}^*$ of the model is the vector that satisfies

$$x_i^* = \sum_{j=0}^{r} T_{ij} x_j^* \tag{4.4}$$

where the entries of transition probability matrix T are given by sampling the heuristic r times. The heuristic for a population where j out of r individuals are genome 0, as in (4.1), is given by

$$\mathcal{G}\left(\left(\frac{j}{r}, 1-\frac{j}{r}\right)^{\top}\right) = \left((1-2\mu)\frac{j}{r}+\mu, 1-(1-2\mu)\frac{j}{r}-\mu\right)^{\top}. \tag{4.5}$$

The probability that this model is sampled such that i out of r individuals are genome 0 gives the transition probability

$$T_{ij} = \binom{r}{i}\left((1-2\mu)\frac{j}{r}+\mu\right)^{i}\left(1-(1-2\mu)\frac{j}{r}-\mu\right)^{r-i}. \tag{4.6}$$

We want to find the conditions for which this equilibrium distribution $\mathbf{x}^*$ is uniform, i.e., when (4.4) is according to

$$1 = \sum_{j=0}^{r}\binom{r}{i}\left((1-2\mu)\frac{j}{r}+\mu\right)^{i}\left(1-(1-2\mu)\frac{j}{r}-\mu\right)^{r-i}. \tag{4.7}$$

By introducing $x = \frac{j}{r}$ and by multiplying both sides with $\frac{1}{r}$, the summation can be replaced by an integral, describing a continuous approximation of the equilibrium distribution with

$$\frac{1}{r} = \binom{r}{i}\int_0^1 \left((1-2\mu)x+\mu\right)^{i}\left(1-(1-2\mu)x-\mu\right)^{r-i} dx \tag{4.8}$$

for $i = 0, 1, \ldots N$. This approximating integral can be obtained by replacing the states of the finite model by a continuous state space, e.g., by using the trapezium rule for integration. The error of this approximation is of order $O(1/r^2)$ and mainly affects the correctness of states with $i = 0$ and $i = r$.

By introducing a new variable $y = (1-2\mu)x+\mu$, where $dy = (1-2\mu)dx$ we can rewrite (4.8) as

$$\frac{1}{r} = \binom{r}{i}\frac{1}{1-2\mu}\int_{\mu}^{1-\mu} y^{i}(1-y)^{r-i}dy. \tag{4.9}$$

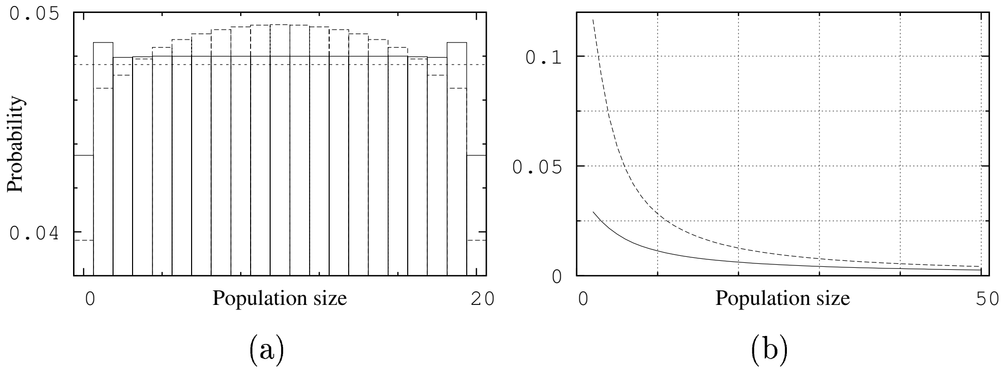


Figure 4.4: (a) Limit distribution of the model for $r = 20, \mu = \frac{1}{2(r+1)}$ (solid lines) and for $r = 20, \mu = \frac{1}{2r}$ (dashed lines) and the uniform distribution (dotted). (b) Euclidean distance from the limit distribution of the model – for various r and $\mu = \frac{1}{2(r+1)}$ (solid) and $\mu = \frac{1}{2r}$ (dashed) – to the uniform distribution.

Function $y^i(1-y)^{r-i}$ has a sharp peak with maximum at $y = \frac{i}{r}$, and tends very fast to 0 if y moves away from this peak. For values of i that are not close to 0 or r and for small mutation rates μ, the sharp peak is not close to the integration borders μ and $1-\mu$, and consequently within the integration interval. Replacing the integration borders with 0 and 1 thus gives a good approximation of the integral in (4.9), for values of i that are not near 0 and r, hence

$$\frac{1}{r} = \left(\begin{array}{c} r \\ i \end{array} \right) \frac{1}{1-2\mu} \int_0^1 y^i (1-y)^{r-i} dy. \tag{4.10}$$

The integral is the beta integral, or Eulerian integral of the first kind (Whittaker & Watson 1990), with value $\frac{i!(r-i)!}{(r+1)!}$ whose substitution yields

$$\begin{aligned} \frac{1}{r} &= \left(\begin{array}{c} r \\ i \end{array} \right) \frac{1}{1-2\mu} \frac{i!(r-i)!}{(r+1)!} \\ &= \frac{r!}{i!(r-i)!} \frac{1}{1-2\mu} \frac{i!(r-i)!}{(r+1)!} \\ &= \frac{1}{(1-2\mu)(r+1)}. \end{aligned} \tag{4.11}$$

Note that all occurrences of i have dissolved. Further simplification of (4.11) gives

$$\mu = \frac{1}{2(r+1)}. \tag{4.12}$$

Our derivation thus expects a limit behavior that can be approximated by the uniform distribution $\mathbf{x}^u$ with $x_i^u = \frac{1}{r+1}$ when the mutation rate μ equals

$\frac{1}{2(r+1)}$, which is only slightly different from Wright's (1931) prediction of $\mu = \frac{1}{2r}$. Because of the approximation in step (4.9-4.10) we expect that these distribution probabilities x_i^u are fairly good approximations, if i is not close to 0 or r.

We can verify this prediction by computing the exact limit behavior of the model for parameters with $\mu = \frac{1}{2r}$ and $\mu = \frac{1}{2(r+1)}$, and comparing these distributions to a uniform distribution $\mathbf{x}^u$ whose probabilities are all equal to $\frac{1}{r+1}$.

Figure 4.4(a) shows the limit distribution of the model with no selection for $r = 20$ with $\mu = \frac{1}{2(r+1)}$ and $\mu = \frac{1}{2r}$, and compares these with a uniform distribution. Note that the limit distribution of our expectation is "flatter" around the middle, and is generally closer to the uniform distribution, thus representing a better expectation of the transition than Wright's expectation at $\mu = \frac{1}{2r}$.

Figure 4.4(b) gives the Euclidean distance $\sqrt{(\mathbf{x}^* - \mathbf{x}^u)^2}$ from the limit distributions $\mathbf{x}^*$ (of both our and Wright's predictions of "uniform" limit distributions) to the uniform distribution $\mathbf{x}^u$ of dimension $r+1$ whose entries are all $\frac{1}{r+1}$. It is clear that the prediction of $\mu = \frac{1}{2(r+1)}$ gives a limit distribution that is closer to the uniform distribution as compared to the limit distribution according to Wright's prediction with $\mu = \frac{1}{2r}$. The mutation rate $\mu = \frac{1}{2(r+1)}$ thus better represents the relation between mutation rate and population size, for the transition from the influence of genetic drift through random sampling on one hand and to mutational pressure on the other hand.

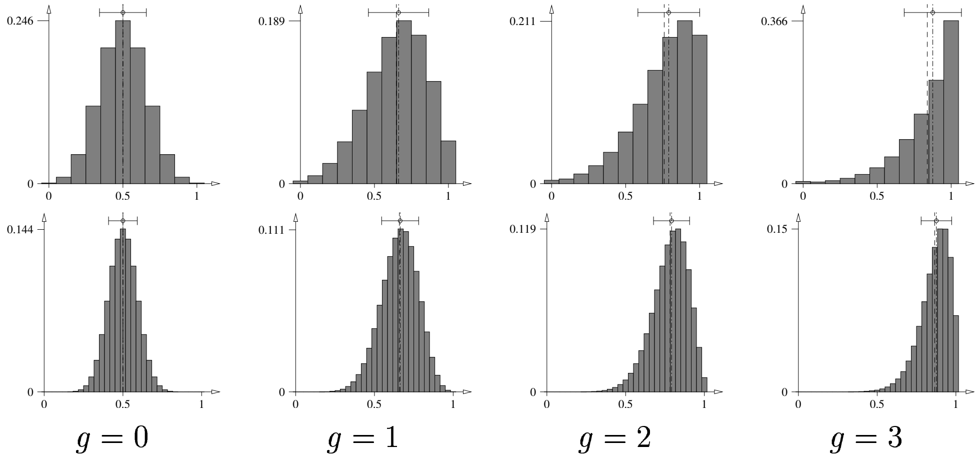
Up until this point, our model did not include selective differences among the genotypes. In the following model, we drop this assumption. One genotype is set to be selectively advantageous over the other, and we study how selective pressure influences the importance of genetic drift.

4.3 One Locus, Two Alleles, Selection

4.3.1 Model

In the following models, we assume static selective pressure. A fitness function f is adopted that is defined by $f(1, P) = 1$ and $f(0, P) = 1 - L$ where $0 < L < 1$. As L becomes larger, the selective pressure is also increased. Parameter L thus represents the amount of selective pressure in the model. In the case that $L = 0$, no selective pressure is exerted on the individuals in the population, as discussed in the previous section. In the model, the mean fitness of a population P equals

$$\begin{aligned} \overline{f}(P) &= p_1 f(1, P) + p_0 f(0, P) \\ &= p_1 1 + (1 - p_1)(1 - L) \\ &= 1 - L(1 - p_1). \end{aligned}$$



$g = 0$ $g = 1$ $g = 2$ $g = 3$

Figure 4.5: Distributions of transient behavior in a haploid model at initial generations $g = 0, 1, 2, 3$ for $r = 10$ (top) and $r = 30$ (bottom) with fitness function $f(0, P) = 0.5, f(1, P) = 1$ and mutation rate $\mu = 0.01$. The horizontal axis represents states by their proportion of 1 individuals, the vertical axis represents the probability of being in a certain state. The infinite model is represented by a dash-dotted vertical line and the mean of the distribution of the finite model is represented with a dashed vertical line. The horizontal error bars represent the expected deviation of the finite model with respect to the infinite model.

The probabilities that genomes 0 and 1 are selected from the population are given by

$$\begin{aligned} S(0, P) &= \frac{p_0 f(0, P)}{\overline{f}(P)} = \frac{(1 - p_1)(1 - L)}{1 - L(1 - p_1)} \\ S(1, P) &= \frac{p_1 f(1, P)}{\overline{f}(P)} = \frac{p_1}{1 - L(1 - p_1)}. \end{aligned}$$

The probability that an individual of type 1 is generated from this selection, according to a "bit flip" mutation rate μ is then equal to

$$\begin{aligned} G(1, P) &= (1 - \mu)S(1, P) + \mu S(0, P) \\ &= \frac{(1 - \mu)p_1 + \mu(1 - p_1)(1 - L)}{1 - L(1 - p_1)} \\ &= \frac{p_1(1 - 2\mu + \mu L) + \mu(1 - L)}{1 - L(1 - p_1)}. \end{aligned}$$

The heuristic function $\mathcal{G}$ for this model is given by these probabilities with

$$\mathcal{G}(\mathbf{p}) = (1 - G(1, P), G(1, P))^{\top}. \tag{4.13}$$

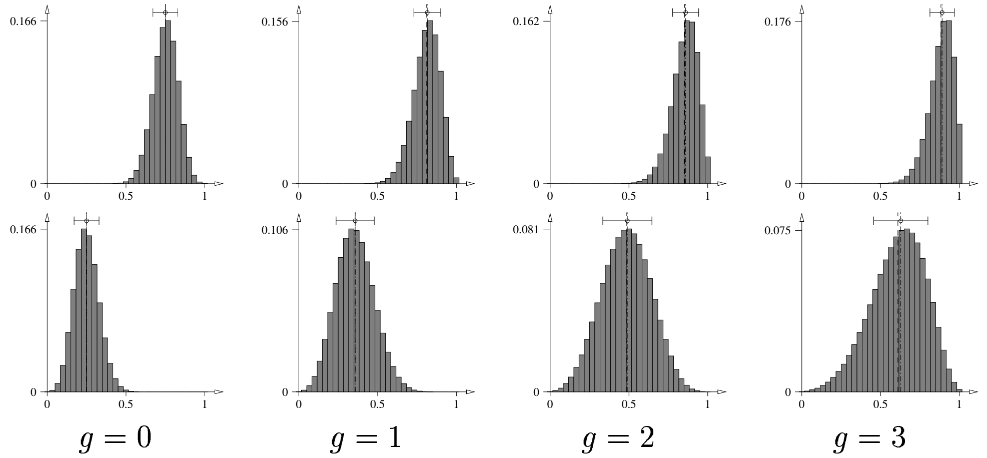


Figure 4.6: Phenotypically mapped distributions of transient behavior in a diploid model at initial generations $g = 0, 1, 2, 3$ for $r = 30$ with fitness function $f(0, P) = 0.5, f(1, P) = 1$ and mutation rate $\mu = 0.01$. The horizontal axis represents populations according to their proportion of phenotype 1. In the top distributions, advantageous allele 1 dominates deleterious allele 0, and in the bottom distribution, deleterious allele 0 dominates advantageous allele 1.

The fixed points $\mathbf{p}^*$ of heuristic $\mathcal{G}$ are given by their component p_1^*

$$p_1^* \;=\; \frac{p_1^*(1 - 2\mu + \mu L) + \mu(1 - L)}{1 - L(1 - p_1^*)}$$

and $p_0^* = 1 - p_1^*$. For a given selective pressure L and mutation rate μ we can now determine the fixed point in Λ of the heuristic $\mathcal{G}$. This fixed point gives the expected proportion of genome 0 in the limit behavior of the infinite population model.

For the finite model with population size r, the heuristic function $\mathcal{G}$ is sampled r times. We construct the Markov model with transition probability matrix T according to (3.21). This matrix is used for studying the behavior of the finite population model.

4.3.2 Transient behavior

Haploid model. The distributions of the transient behavior of the haploid model for the first few generations for $r = 10$ and $r = 30$ are depicted in Figure 4.5, both with $L = 0.5$ (i.e., the fitness of individuals of type 1 is twice that of individuals of type 0) and $\mu = 0.01$. The initial distributions at $g = 0$ are given as in (3.28).

As consecutive generations are formed, the probability of having a population with higher proportions of 1s increases. The expected deviation with respect to the infinite model changes over time as evolution progresses. Initially, the population searches for regions in the search space with high fitness, causing an increase in the standard deviation of the distribution about the mean of the finite

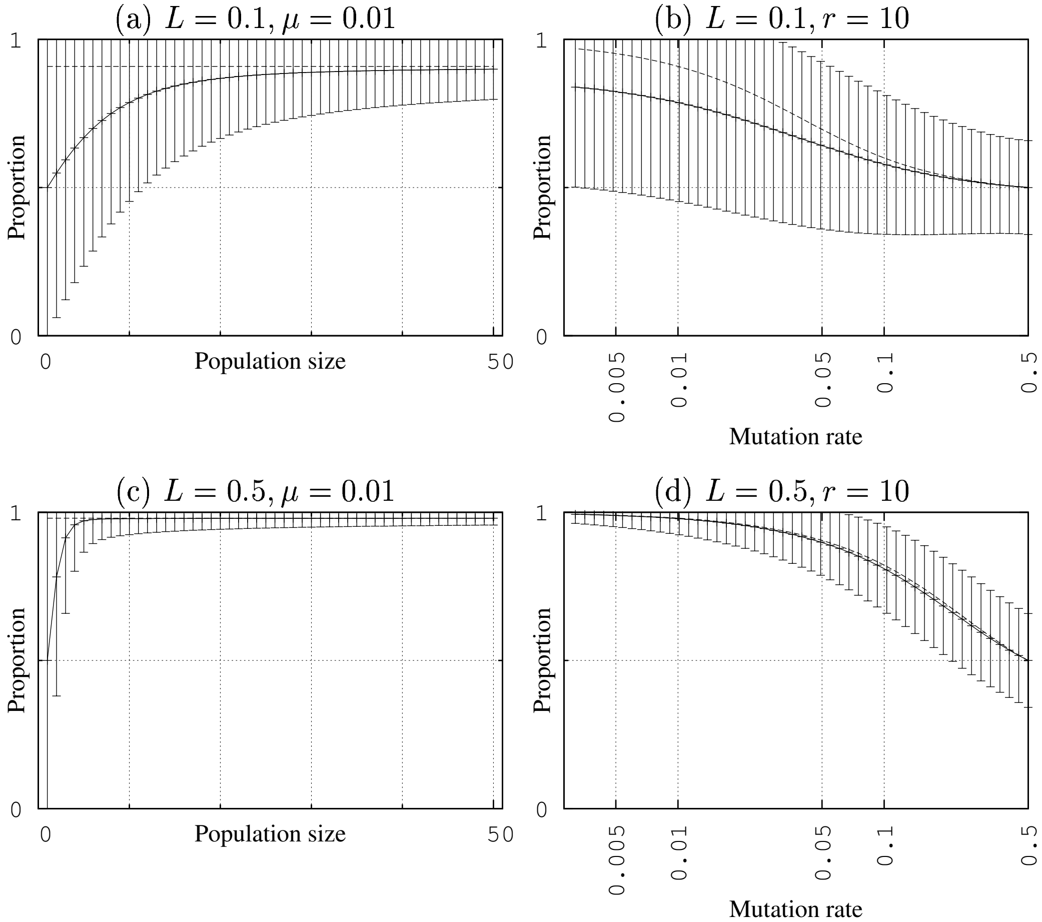


Figure 4.7: Expected proportion of 1 genomes and standard deviation in the limit distribution of the haploid system with selective pressure (a,b) $L = 0.1$ and (c,d) $L = 0.5$, for (a,c) different population sizes and (b,d) different mutation rates. The graph in dashed line type represents the expected proportion of allele 1 in the corresponding model with infinite population size.

model. Also, as we discuss later on, the search in the finite model progresses more slowly than the progress made in the infinite model. Since the expected proportions differ from the proportions in the infinite model, this difference results in a higher expected deviation of the finite model, as discussed in Section 3.7.2.

In the search process, the system then focuses on populations with high mean fitness. The population "converges" to states with a high proportion of highly fit individuals, again becoming more similar to the expectation of the infinite model, resulting in a gradual decrease of the expected deviation. For generations $g > 0$, the mean of the distribution of the finite model differs slightly from the population in the infinite model. We discuss these deviations among expected proportions of alleles between the finite and infinite population models, and the origin thereof in Section 4.3.3.

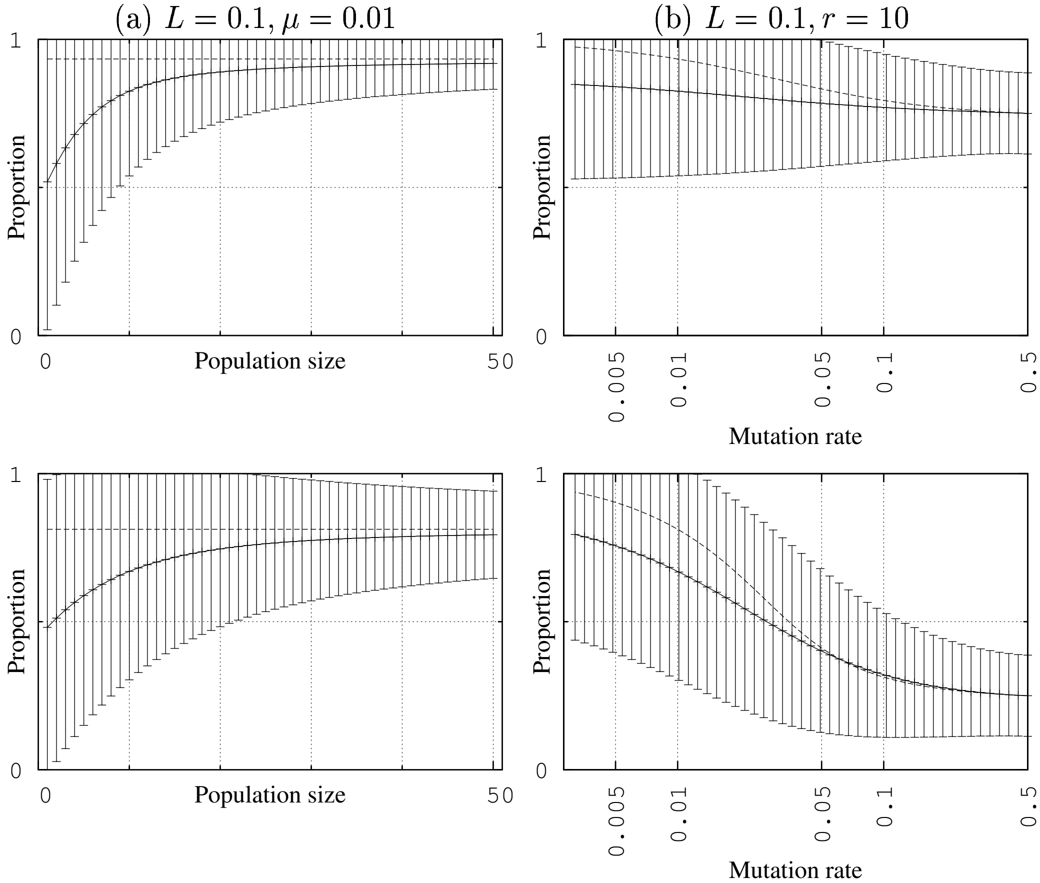


Figure 4.8: Expected proportion of phenotype 1 and its standard deviation in the limit distribution of the diploid system with selective pressure $L = 0.1$ for (a) different population sizes and (b) different mutation rates. In the top figures, advantageous allele 1 is dominant, in the bottom figures deleterious allele 0 is dominant.

Diploid model. Figure 4.6 depicts the distributions, after phenotypical mapping through δ, as in Example 3.2, of the first generations of the corresponding diploid model. Similarly to the haploid model, the probability of having a population with high proportion of phenotype 1 increases over time, but slower.

If the fitter allele is also the dominant allele, the diploid algorithm produces better proportions of the fit allele in comparison with the haploid model. Vice versa, if the fit allele is recessive, the results are worse than the haploid model. If allele 1 is dominant, both homozygous and heterozygous individuals that are represented by the genotypes $\{1, 1\}$ and $\{0, 1\}$ have a high fitness, with only the homozygous individuals with genotype $\{0, 0\}$ receiving a low fitness. In contrast, if allele 0 is dominant, the heterozygous individuals also have a lower fitness. Thus, if the dominant allele has a higher fitness, its phenotypic expression through the dominance operator gives the population a benefit, resulting in a better performance in the system where the fitter allele is dominant over the other. It's thus easier for the population with dominant allele 1 to find the populations with high fitness, as compared to the population whose allele 1 is recessive. Another notable effect of dominance, is that the standard deviation in the model with the less fit dominant allele is higher than the model where the fit allele dominates. This is because – in the model where the fit allele is recessive – unfit heterozygous individuals are generated with a higher probability, even if only a small proportion of the population contains unfit dominant alleles, increasing the diversity in fitness of the populations in the distribution, which consequently results in a distribution with a higher standard deviation.

4.3.3 Limit behavior

Figures 4.7 and 4.8 show the influences of population size r, mutation rate μ, selective pressure L and ploidy on the expected proportions of alleles in the long run. The limit distribution is the unique eigenvector $\mathbf{x}^*$ with eigenvalue 1 of the respective transition matrix.

In contrast with the model without selective pressure, the expected proportions of 0 alleles (or 1 alleles) in the finite model's limit distribution deviate from the expectations of the infinite population model. This is due to the stochastic effects of random sampling, and thus of genetic drift acting upon the systems. Indeed, the system balances the processes of selection and mutation – as predicted by the infinite population model – on one hand and the process of random sampling – which results in genetic drift – on the other hand. The deviation of expected proportions of alleles in the finite model's limit behavior with respect to the infinite population model increases as the mutation rate, population size or selective pressure decreases. These parameters weaken the effectiveness of selection and mutation processes in comparison with the influence of random sampling, and thus drift. Genetic drift is a random process, and is not focused on generating higher proportions of the fitter allele, but works symmetrically for both alleles. Consequently, the expectations of fitness in a deterministic infinite population model may serve as an upper bound of the expected fitness of the finite population model.

We discuss the specific influence of the different parameters in the model separately in the following paragraphs.

Selective pressure. If selective pressure is less prominent, the deviations of the finite population with respect to the infinite population model are more significant. Indeed, as the selective pressure is decreased, the direction in which populations evolve toward higher proportions of fit individuals becomes less focused. This is also the case in the infinite model, where the expected proportion of advantageous alleles in the limit behavior is lower as the selective pressure is decreased.

Accordingly, the influence of genetic drift on the finite model becomes more important, as compared to the influence of selecting individuals with the advantageous allele. In the extreme, where no selective pressure is present, i.e., when $L = 0$, genetic drift and variation are the only processes governing the dynamics of the system, as in the previous system of Section 4.2.

As the selective pressure is increased, the infinite model is more focused on acquiring higher proportions of fit individuals. As a result, drift has less effect on the process in the finite population model, and the expected behavior of the finite model then also better resembles that of the infinite model.

Population size. Since more stochastic effects of random sampling are expected in smaller populations, as in the model with no selective pressure, the deviation between a finite population's limit distribution and the expected population of the infinite model grows as the population size in the finite population model is diminished. Smaller populations tend to perform worse with regard to fitness than bigger populations, as the sampling allows deleterious alleles to be randomly produced in the population at the next generation, as compared with the generations in the infinite model. In the extreme case that $r = 1$, the limit distribution of the system has probability $1/2$ for either state, with either alleles 0 or 1 being fixated. This observation can be made independent of the selective pressure, as in the model where populations only contain one individual, there is no selection. If $r = 1$, the selection process can only render the one individual in the population, independent of its fitness. Thus, as the population size is decreased to this extreme where $r = 1$, the limit distribution converges to this behavior where either allele has an equal probability of ending up in the limit distribution.

This positive relationship between population size and performance of the system has been commonly observed by practitioners of genetic algorithms, where large populations are said to have more computational power at hand to find a good solution for an optimization problem. The observed decrease in computational power of a small population is due to the increased influence of stochastic effects of random sampling at each generation. Because of random sampling, individuals with deleterious alleles have a greater probability of invading these small populations. The loss in computational power of small populations can thus be explained in terms of genetic drift within the finite population model.

Mutation rate. As the selective pressure is low, the finite model deviates more from the expected population of the infinite population model as the mutation rate is decreased. This is also due to the balance between the infinite model's dynamics and drift.

We have observed – in the system where no selection is present – that the standard deviation of the finite model's limit behavior increased as the mutation rate becomes smaller, i.e., genetic drift and the fixation of alleles became more prominent as the rate of mutation was too slow to introduce new alleles to the population. With a slow mutation rate, alleles thus have a higher probability of becoming fixated, and others become lost from the population, as new mutant alleles have a slow rate of entering and invading the population.

The model with selective pressure finds a balance between two forces. On one hand, there's the dynamics of the infinite population model which is governed by the selective and variational pressure. On the other hand, genetic drift in the finite model forces the population towards less diverse populations. If the selective pressure is sufficiently low, the expected population in the finite model deviates from the expected population in the infinite model as the mutation rate is decreased. If the selective pressure is sufficiently high, the selection process can undo the effects of random sampling.

Ploidy and dominance. The influences of the parameters in the previous sections also apply to the expectations of the diploid model. In coherence with the observations of the transient behavior of the diploid model, better proportions of fit individuals are expected if the dominant allele is the fitter one, even better than the expectations of the haploid model. It should however be pointed out it may be observed – from simulations and the implementation of the power method – that the time required to hit the steady state distribution of the diploid model is longer than the time to reach the steady state for the haploid model. We have however not studied this observation as our main focus is the long term behavior of the systems.

Summary

We have adopted the models of the previous chapter to indicate how the parameters of the model influence the long term behavior in evolutionary systems in environments that remain static over time. We have observed that, for sufficiently small populations, or sufficiently low mutation rates, that the system is mainly governed by drift. There is, consequently, a higher probability of fixation with these parameter settings. For sufficiently large population sizes and fast mutation rates, the long term behavior of the evolutionary system has a higher probability to end up in populations whose composition of individuals is highly diverse. We have provided a correction of Wright's prediction on the boundary between drift and mutation. If selection is assumed, these effects are also at work, but the model finds a balance between the forces of selection and mutation on one hand, and fixation of alleles because of drift on the other hand. Drift becomes stronger as the selective pressure in the population is decreased. Dependent on the dominance of the fit allele, a diploid population may perform better in the long run as compared to a haploid population.

Relation to genetic algorithms. In genetic algorithms, larger population sizes do not necessarily result in better run times of implementations of genetic algorithms, as the fitnesses of the individuals of a genetic algorithm are not computed in parallel. Instead, a generation in an implementation of a genetic algorithm usually requires $O(r)$ fitness evaluations, as all of the individuals have to be evaluated by a – usually costly – fitness function. The results given in this section discuss the behavior of the genetic algorithms in the long run, so we have ignored these implementation details in this study. However, even in the long run, the effects of a population size can be observed, and are thus not only of importance for the transient behavior of a genetic algorithm. For a specific study of the transient behavior of genetic algorithms in function optimization with finite Markov models, see De Jong (1995).

The observations of choosing a small mutation rate as being optimal in the long run behavior are also in contrast with the experiences of a practitioner in genetic algorithms. In the long run, a low mutation rate is preferred as it does not allow any deleterious mutant alleles to enter the population, if it has a high proportion of advantageous alleles. When studying the transient behavior of a genetic algorithm, however, setting the mutation rate too low also refuses to allow advantageous mutants to enter a population of deleterious alleles. In a practical implementation of a genetic algorithm, the mutation rate must thus be chosen such that it balances the exploration of new areas of the search space, which may contain promising advantageous mutants, and the exploitation of these advantageous areas of the search space, by not setting the mutation rate too high. As the mutation rate of a model is higher, the resulting Markov chain mixes more rapidly. In that case the steady state of the Markov chain is reached more quickly, i.e., the behavior of the genetic algorithm converges faster to its limit distribution. However, as we can see from the observations of the long run behavior of the model, increasing the mutation rate leads to smaller probabilities of obtaining highly fit individuals in the long run. See Vitányi (2000) which discusses the relation of this balance for the design of *good* genetic algorithms.

In the short run, there is no obvious benefit of adopting diploid populations over haploid populations for function optimization in genetic algorithms, but better performance may expected in the long run, dependent on the dominance scheme. We have not considered models where the dominance of alleles can change over time, but it is clear that such schemes may be beneficial for the performance of a diploid genetic algorithm, both in the short and long run.

Inherently with the transient behavior, there is a higher standard deviation in the limit distribution in the diploid model as compared to the haploid model. A diploid population thus maintains more diversity throughout its transient evolution, which may turn out to be beneficial once we study dynamically changing fitness environments in the following chapters.

Chapter 5

Explicitly Dynamic Fitness Environments

We construct Markov models of evolutionary systems in environments whose fitness function changes over time. Two separate models are presented, with fitness functions that are alternated deterministically or stochastically over time. In the case of ergodicity, the limit behavior of these systems can be utilized to express predictions of expected behavior and measurements of performance for the algorithm and its parameters. We provide methods to study the limit behavior and performance of the evolutionary systems in their dynamic environments. We also show how the stochastic and deterministic environment models can be applied to study the influence of the system's parameters – ploidy, mutation rate, frequency of changes in the environment, population size and selective pressure – on the long run performance in the respective environments. We compare these observations to the conclusions of the previous chapter.

We shortly introduce dynamic optimization problems, and construct Markov models of evolutionary systems in deterministically and stochastically alternating fitness functions. A study of the expected behavior of the evolutionary system in an environment with an alternating deleterious allele is provided.

The contents of this chapter is based on two published papers, namely A.M.L. Liekens, H.M.M. ten Eikelder, P.A.J. Hilbers, *Finite Population Models of Dynamic Optimization with Alternating Fitness Functions* (2003c) and A.M.L. Liekens, H.M.M. ten Eikelder, P.A.J. Hilbers, *Finite Population Models of Dynamic Optimization with Stochastically Alternating Fitness Functions* (2003d).

5.1 Introduction

Genetic algorithms commonly address optimization problems that are static in time, as in the previous chapter. Similarly, most applications of studies in population genetics focus their research on environments in which selection is static over time. Many real world applications of optimization techniques, and real world natural environments, have time dependent characteristics. In dynamic optimization, online optimization techniques try to track optima of changing problems (Branke 2001). Branke (2003) gives an overview of adaptations of genetic algorithms to better track changes in optimization problems and their optima.

Models of dynamic optimization. Models of populations with infinite population size in dynamical environments have been studied in (Rowe 2001) and (Ronnewinkel, Wilke, & Martinetz 2001). Dynamics of evolutionary techniques for the dynamic bit matching problem have been studied by Droste (2002) and Branke & Wang (2003). Arnold & Beyer (2002) study the behavior of an Evolution Strategy tracking the optimum of a moving peak problem.

In this chapter, we extend the models of Chapter 3 for dynamic environments. Our approach is to combine several probability transition matrices of Markov models of evolutionary systems, each with a specific static fitness function, into a new probability transition matrix describing the evolutionary system in its dynamic environment. Two approaches for the alternation of fitness functions – a deterministic and a stochastic interpretation of changes in the environment – are proposed.

Deterministic alternation of fitness functions. We first present a stochastic model with finite populations in a deterministically dynamic environment. In this model, a fitness function governs the selection of the population for a given number of generations, after which another fitness function takes over the selection process. After such a number of fitness functions have governed the selection for a fixed number of generations each, the alternation is restarted with the first fitness function. Stochastic transition matrices of consecutive generations, with their distinct fitness function, are combined into one Markov matrix that gives the transitions between the states for a complete cycle of the environment. We can then determine and analyze the limit behavior of these stochastic systems. In order to find expectations of performance of the evolutionary system toward the limit, we unfold the combined matrix and its eigenvector to calculate an expectation of fitness, based on the limit behavior of the combined chain.

The applications of this first model are rather artificial because of determinism in the dynamic environments. Applications of dynamic problems for genetic algorithms practitioners and dynamic environments for population geneticists are rather stochastic in nature.

Stochastic alternation of fitness functions. A second model is constructed that better serves the study of evolving populations in these stochastic dynamic applications. This second model adopts a Markov chain to model the stochastic environment itself. In this Markov model of the environment, each fitness function is considered as a state, and transition probabilities among these states define the stochastic dynamics of the environment. Steps in this changing environment and steps in the evolutionary model – according to the current state of the environment – are alternated. We construct a Markov model that combines the stochastic changes in the environment and in the population. We adopt such models to study the limit behavior and performance of the evolutionary model, with respect to its stochastically changing fitness environment.

5.2 Deterministically Alternating Fitness Functions

5.2.1 Model

We consider an optimization problem for which the fitness function $f(s,P,g)$ – for a solution $s \in \Omega_S$ and population P – changes with time g. In this first model, the dynamics of the fitness function is deterministic and periodic with a period of g_{tot} generations, i.e., the fitness at time $g + g_{tot}$ is equal to the fitness at time g. In this cycle, a deterministic chain of fitness functions governs the selection in the evolutionary system. We let each of the fitness functions be active for a number of generations, summing up to g_{tot} generations.

Consider an evolutionary system with its parameters and the set of n static fitness functions $F = \{f_1, f_2, \ldots, f_n\}$. Let τ_i describe the state transitions of the evolutionary model, with selection according to fitness function f_i. Let T_i denote the Markov matrix of the system according to transitions τ_i. We assume that all other parameters of the model – such as population sizes and parameters of reproduction – are equal for any of the n evolutionary systems. We assume that during a run of the model each of the n fitness functions f_i governs the selection alternately for a fixed finite number of generations g_i, with $g_{tot} = \sum_{i=1}^{n} g_i$, such that the fitness function at time g for a solution $s \in \Omega_S$ in a population P is given by

$$f(s,P,g) = f_i(s,P) \tag{5.1}$$

where i satisfies

$$\sum_{j=1}^{i-1} g_j \leq (g \mod g_{tot}) < \sum_{j=1}^{i} g_j \tag{5.2}$$

Consequently, we can construct a probability transition matrix T_{det} with

$$T_{det} = T_n^{g_n} \ldots T_2^{g_2} T_1^{g_1}. \tag{5.3}$$

This transition matrix T_{det} gives the composited transition of the evolutionary system for g_{tot} generations, starting the initial (0th) generation with fitness function f_1, and ending with the last generation of fitness function f_n. Consequently, a run of the model repeatedly visits all fitness functions and simulates a dynamic environment.

5.2.2 Limit behavior

Existence of a unique limit. In Appendix B.3, we show that the combination of irreducible and aperiodic Markov matrices $T_1, \ldots, T_n$, as defined above, does not always result in a transition matrix T_{det} that is irreducible and aperiodic. Therefore, we cannot simply assume that the Markov chain based on transition matrix T_{det} converges to a unique equilibrium distribution.

We can, however, make the following assumption. If mutation can map any individual to any other individual in the algorithm's search space with a strictly positive probability – i.e., when the mutation rate is according to $0 < \mu < 1$ – then all elements in transition matrices T_i are strictly positive, see Theorem 3.2. Consequently, all transition probabilities of the combined model T_{det} in (5.3) are strictly positive. This makes the combined Markov model ergodic, and hence, due to Perron-Frobenius, there exists a unique stochastic eigenvector of the matrix with corresponding eigenvalue 1. This eigenvector describes the fixed point distribution of the system toward the limit. The distribution denotes the probabilities of being in a state of the system, at the start of a cycle of the fitness functions.

Interpretation of the limit. Let $\mathbf{x}^*_{(0)}$ denote the unique eigenvector, with corresponding eigenvalue 1, of the irreducible and aperiodic transition matrix T_{det}. By definition, $\mathbf{x}^*_{(0)} = T_{det}\mathbf{x}^*_{(0)}$ or more specifically, $\mathbf{x}^*_{(0)} = T_n^{g_n} \dots T_2^{g_2} T_1^{g_1} \mathbf{x}^*_{(0)}$. The eigenvector describes the probability distribution over the states of the system, at the start of the environment's cycle. Let $\mathbf{x}^*_{(u)}$ denote the distribution over all states of the system, u generations since eigenvector $\mathbf{x}^*_{(0)}$, with $0 \leq u \leq g_{tot}$, i.e.,

$$\mathbf{x}^*_{(u)} = T_v^w T_{v-1}^{g_{v-1}} \dots T_1^{g_1} \mathbf{x}^*_{(0)} \tag{5.4}$$

where $u = g_1 + g_2 + \cdots + g_{v-1} + w$ and $0 \leq w < g_v$. All $\mathbf{x}^*_{(u)}$ are vectors describing the consecutive distributions over the state space with $\mathbf{x}^*_{(0)} = \mathbf{x}^*_{(g_{tot})}$.

5.2.3 Expected performance

Since we know the fitness function at each of the generations, we can compute the expected fitness over all generations, as the system defined by T_{det} converges toward the limit. This mean fitness, derived from the exact eigenvector, gives us the expected mean fitness of a simulation run of the system.

Let $\left(\mathbf{x}^*_{(u)}\right)_P$ denote the probability of being in state P, at u generations since eigenvector $\mathbf{x}^*_{(0)}$. Let $f_{(u)}$ denote the fitness function that is applied at the uth generation since $\mathbf{x}^*_{(0)}$. Let $\overline{f_{(u)}}$ denote the weighted mean fitness of the populations according to distribution $\mathbf{x}^*_{(u)}$ with

$$\overline{f_{(u)}} = \sum_{P \in \pi} \left(\mathbf{x}^*_{(u)}\right)_P \overline{f_{(u)}(P)} \tag{5.5}$$

where $\overline{f_{(u)}(P)}$ denotes the mean fitness of population P according to fitness function $f_{(u)}$. The overall mean fitness $\overline{f}$ of all $\overline{f_{(u)}}$ with

$$\overline{f} = \frac{1}{g_{tot}} \sum_{u=1}^{g_{tot}} \overline{f_{(u)}} \tag{5.6}$$

gives the expected fitness over all generations, as the system goes toward the limit.

5.3 Stochastically Alternating Fitness Functions

5.3.1 Introduction

Similarly to the model with deterministically alternating fitness functions, let $F = \{f_1, f_2, \ldots, f_n\}$ denote a set of n fitness functions. Instead of studying an evolutionary system that is based on selection with a deterministic chain over these fitness functions, we now stochastically step through this set of fitness functions as the algorithm proceeds. This results in a dynamically changing fitness environment for the optimizing algorithm. We use a Markov chain to define the transition probabilities $E_{j,i}$ in the environment. If the selection of the system is currently based on fitness function f_i, then $E_{j,i}$ denotes the probability that the fitness function used at the next generation is f_j. The stochastic matrix E with elements $E_{j,i}$ gives the Markov matrix defining the dynamical behavior of our stochastic model of a changing environment. We assume that the environment's matrix E is irreducible and aperiodic, such that the environment's limit behavior is given by E's unique stochastic eigenvector with eigenvalue 1, and thus independent of the initial distribution over the environment's states.

In the following section, we discuss how this stochastic environment model can be combined with a stochastic model of the evolutionary systems as in the previous sections to study the behavior of evolutionary systems in stochastically dynamic environments.

5.3.2 State space

In order to represent each possible state of the evolutionary system in its stochastic environment, we need to enumerate all states of the evolutionary model, i.e., the set of possible populations π and the set of possible fitness functions, $F = \{f_1, f_2, \ldots, f_n\}$. The number of possible populations and number of states is assumed to be finite. The state space of the whole system in its environment can be represented as the Cartesian product of these two sets, i.e., $F \times \pi$. A state $(f_i, P) \in F \times \pi$ comprises of a population P and fitness function f_i.

Let $\mathbf{x}$ represent a stochastic distribution vector over this state space, such that $x_{(f_i,P)}$ represents the probability of being in state (f_i, P). For notational purposes, we can rewrite $\mathbf{x}$ as $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)^\top$ where each sub vector $\mathbf{x}_i$ denotes the distribution over π for fixed fitness function f_i. Note that, since $\mathbf{x}$ is a stochastic vector, the elements in sub vectors $\mathbf{x}_i$ have sum $\sum_{P\in\pi} x_{(f_i,P)} \leq 1$ which denotes the probability of being in a state with fitness function f_i, according to distribution $\mathbf{x}$.

5.3.3 Transitions

Generation step. Consider matrices $T_1, T_2, \ldots, T_n$ to be n transition matrices of a specific evolutionary model and its parameters, with selection proportional to n different fitness functions $f_1, f_2, \ldots, f_n$, respectively. We assume that these transition matrices are irreducible and aperiodic. Let $\mathbf{T}$ be the block diagonal matrix with diagonal elements T_i, i.e.,

$$\mathbf{T} = \begin{pmatrix} T_1 & 0 & \cdots & 0 \\ 0 & T_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & T_n \end{pmatrix}. \tag{5.7}$$

We let $|\pi| = |T_1| = |T_2| = \ldots = |T_n|$ denote the number of rows or columns of the transition matrices. Consequently, $\mathbf{T}$ is a $n|\pi| \times n|\pi|$ matrix.

If $\mathbf{x}$ denotes a distribution over all states of the system in its dynamic environment, then $\mathbf{Tx}$ denotes the distribution after one generation of the evolutionary algorithm. Note that this product evaluates the next generation for all evolutionary systems in their respective fixed fitness functions.

If the initial distribution vector $\mathbf{x}$ only contains states that assume a specific fitness function f_i, then distribution $\mathbf{T}^k\mathbf{x}$ at the kth generation can only have strictly positive elements if these represent states that also belong to fitness function f_i. There is no path possible from any state with one fitness function, to any other state with a different fitness function. Hence, matrix $\mathbf{T}$ is not irreducible. Note that this property holds for any block diagonal matrix.

As transition matrix $\mathbf{T}$ is not irreducible, there is not a unique fixed point that can be used to study the limit – or expected – behavior of the system according to $\mathbf{T}$, due to the Perron-Frobenius theorem. On its own, $\mathbf{T}$ doesn't contain a dynamic environment, which is introduced in the following section.

Environment step. Let $\mathbf{E}$ be the block matrix with

$$\mathbf{E} \;=\; \begin{pmatrix} E_{1,1}I & \cdots & E_{1,n}I \\ \vdots & \ddots & \vdots \\ E_{n,1}I & \cdots & E_{n,n}I \end{pmatrix} \tag{5.8}$$

where I denotes the $|\pi| \times |\pi|$ identity matrix. This matrix has, in similarity to $\mathbf{T}$, $n|\pi| \times n|\pi|$ elements.

If $\mathbf{x}$ denotes a distribution over all states, then $\mathbf{Ex}$ denotes the distribution after one stochastic step in the environment. Distributions over population states for each fitness function are redistributed according to the transition probabilities in the environment's transition matrix E.

Similar to the non-irreducibility argument for matrix $\mathbf{T}$, we can show that the repeated multiplication of $\mathbf{E}$ with an initial vector $\mathbf{x}$ does not yield elements with strictly positive probability, if these states constitute a population that was not present in the initial distribution $\mathbf{x}$. Indeed, the rows and columns of $\mathbf{E}$ can be rearranged – using simultaneous row and column exchanges only – such that the resulting matrix $\mathbf{E}'$ of the form

$$\mathbf{E}' \;=\; \begin{pmatrix} E & 0 & \cdots & 0 \\ 0 & E & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & E \end{pmatrix} \tag{5.9}$$

is a block diagonal matrix. Since matrix permutations do not have an influence on the irreducibility of a transition matrix, $\mathbf{E}'$ and $\mathbf{E}$ are not irreducible.

Combined transitions. The evolutionary system, tracking the dynamic problem defined by the set of fitness functions F and transitions E over this set can now be modeled as a Markov chain with transition matrix

$$T_{sto} = \mathbf{TE} = \begin{pmatrix} E_{1,1}T_1 & \cdots & E_{1,n}T_1 \\ \vdots & \ddots & \vdots \\ E_{n,1}T_n & \cdots & E_{n,n}T_n \end{pmatrix}. \tag{5.10}$$

If $\mathbf{x}$ denotes a distribution over all states, then $\mathbf{TEx}$ denotes the distribution after one stochastic step in the environment, followed by a generational step of the system. We are now interested in the limit and fixed point behavior of this model, which represents an evolutionary system in a stochastically changing environment.

5.3.4 Limit behavior

We assume that $0 < \mu < 1$, such that all elements in matrices T_i are strictly positive according to Theorem 3.2. Although both $\mathbf{T}$ and $\mathbf{E}$ are not irreducible and not aperiodic, we can show that their product $T_{sto} = \mathbf{TE}$ is irreducible and aperiodic as stated by the following theorem:

Theorem 5.1. *Let E represent an irreducible and aperiodic $n \times n$ matrix. Let $T_1, \ldots, T_n$ be n equally sized stochastic matrices with strictly positive elements. Let $\mathbf{T}$ and $\mathbf{E}$ denote the matrices as defined by (5.7) and (5.8). Matrix product $\mathbf{TE}$ is irreducible and aperiodic.*

Proof. We first show that the matrix is irreducible. We then use a similar argument to show that the matrix is aperiodic.

For irreducibility, we have to show that any state (f_j, Q) can be reached from any other state (f_i, P) by moving through consecutive states according to strictly positive probabilities in stochastic matrix $\mathbf{TE}$. Starting at state (f_i, P) we can step through different fitness functions, maintaining the same population, to reach (f_j, P), using transition probabilities from $\mathbf{E}$, since E is irreducible and any diagonal element of any T_k is strictly positive. Any such step, e.g., from (f_a, P) to (f_b, P) is possible with $\mathbf{TE}$ if $E_{b,a}$ is strictly positive.

At the same time, we need the population to move from P to Q. Since all transition matrices T_k have strictly positive elements, we can, at each one of the steps from one fitness function to another, also step from one population to any other. By doing so at one step while moving from f_i to f_j, we can also switch from population P to Q, and thus reach (f_j, Q) from (f_i, P). Since this is possible for any pair of states, $\mathbf{TE}$ is irreducible.

Since E is aperiodic, there exists an infinite set S of lengths of paths going from f_i to f_j such that $\gcd(S) = 1$. Starting in state (f_i, P), we can follow any of these paths as indicated by the irreducibility argument to reach (f_j, Q). The set of path lengths of these possible paths from state (f_i, P) to (f_j, Q) are represented by the same S, with $\gcd(S) = 1$. As a result, $\mathbf{TE}$ is irreducible and aperiodic. □

Because of the Perron-Frobenius theorem, the unique fixed point distribution of $T_{sto} = \mathbf{TE}$ is represented by the unique eigenvector $\mathbf{x}^*$, with corresponding eigenvalue 1, of matrix $T_{sto} = \mathbf{TE}$. We now give some computational considerations to simplify the calculation of this eigenvector.

Implementation of the power method. In the power method, see Algorithm 3.2, repetitive multiplication of an initially distributed stochastic vector $\mathbf{x}(0)$ with T_{sto} converges to the eigenvector with eigenvalue 1. To simplify the computation for T_{sto}, we apply some algebraic manipulation in the definition of eigenvector

$\mathbf{x}^* = T_{sto}\mathbf{x}^*$ with

$$\begin{pmatrix} \mathbf{x}_1^* \\ \vdots \\ \mathbf{x}_n^* \end{pmatrix} = \mathbf{TE} \begin{pmatrix} \mathbf{x}_1^* \\ \vdots \\ \mathbf{x}_n^* \end{pmatrix} \tag{5.11}$$

$$= \mathbf{T} \begin{pmatrix} E_{1,1}\mathbf{x}_1^* + \cdots + E_{1,n}\mathbf{x}_n^* \\ \vdots \\ E_{n,1}\mathbf{x}_1^* + \cdots + E_{n,n}\mathbf{x}_n^* \end{pmatrix}$$

$$= \begin{pmatrix} T_1\left(E_{1,1}\mathbf{x}_1^* + \cdots + E_{1,n}\mathbf{x}_n^*\right) \\ \vdots \\ T_n\left(E_{n,1}\mathbf{x}_1^* + \cdots + E_{n,n}\mathbf{x}_n^*\right) \end{pmatrix} \tag{5.12}$$

so we can find the eigenvector with corresponding eigenvalue 1 of T_{sto} by iterating according to the system of equations defined by

$$\left\{ \begin{array}{rcl} \mathbf{x}_1' & = & T_1\left(E_{1,1}\mathbf{x}_1 + \cdots + E_{1,n}\mathbf{x}_n\right) \\ & \vdots & \\ \mathbf{x}_n' & = & T_n\left(E_{n,1}\mathbf{x}_1 + \cdots + E_{n,n}\mathbf{x}_n\right) \end{array} \right. . \tag{5.13}$$

Note that because of the algebraic transformation from (5.11) to (5.13) we no longer need to store big matrix $\mathbf{TE}$ in the computer's memory when running the power method to find the limit behavior for T_{sto}.

Seeding the initial distribution. The fixed point distribution $\mathbf{x}^*$ of the Markov chain defined by transition matrix $\mathbf{TE}$ is related to eigenvector $\mathbf{e}$ with corresponding eigenvalue 1 of E as stated by the following theorem:

Theorem 5.2. *Let E represent an irreducible and aperiodic $n \times n$ stochastic matrix. Let $\mathbf{e} = (e_1, e_2, \ldots, e_n)^\top$ denote the unique eigenvector, with corresponding eigenvalue 1, of E. Let $T_1, \ldots, T_n$ be n equally sized probability transition matrices with strictly positive elements, defined over state space π. Let $\mathbf{T}$ and $\mathbf{E}$ denote the matrices as defined by (5.7) and (5.8). If $\mathbf{x}^*$ denotes the eigenvector of $\mathbf{TE}$, with corresponding eigenvalue 1, then*

$$\forall i, 1 \leq i \leq n : e_i = \sum_{P \in \pi} x^*_{(f_i,P)} \tag{5.14}$$

Proof. Because of (5.12) we know that sub vector $\mathbf{x}_i^*$ of eigenvector $\mathbf{x}^*$ is given by

$$\mathbf{x}_i^* = T_i\left(E_{i,1}\mathbf{x}_1^* + \cdots + E_{i,n}\mathbf{x}_n^*\right) \tag{5.15}$$

for any $1 \leq i \leq n$. The sum of probabilities of elements in both left and right hand side of (5.15) gives

$$\sum_{P \in \pi} (\mathbf{x}_i^*)_P = \sum_{P \in \pi} \left(T_i \left(E_{i,1}\mathbf{x}_1^* + \cdots + E_{i,n}\mathbf{x}_n^*\right)\right)_P . \tag{5.16}$$

Since all T_i matrices are stochastic, they have no influence on computing the sum of vector elements, such that the right hand side of (5.16) can be reduced, before redistributing the summation, to

$$e_i' = E_{1,i}e_1' + \cdots + E_{n,i}e_n' \tag{5.17}$$

with scalars $e_i' = \sum_{P\in\pi} (\mathbf{x}_i^*)_P = \sum_{P\in\pi} x^*_{(f_i,P)}$. We let $\mathbf{e}'$ denote a vector with $\mathbf{e}' = (e_1', e_2', \ldots, e_n')^\top$. Equation (5.17) can now be rewritten as $\mathbf{e}' = E\mathbf{e}'$, thus eigenvector $\mathbf{e}$ with corresponding eigenvalue 1 of E equals $\mathbf{e}'$, with the result that

$$e_i = e_i' = \sum_{P\in\pi} x^*_{(f_i,P)}$$

holds for all $1 \leq i \leq n$. □

Note that this property of the eigenvector is due to the transitions in E, which occur independently from the rest of the system as defined by $\mathbf{TE}$.

Since the eigenvector with corresponding eigenvalue 1 of E can be computed very easily, as compared with computing the eigenvector of $\mathbf{TE}$, the initial distribution $\mathbf{x}(0)$ for iterating system (5.13) can be seeded as indicated by the theorem, according to vector $\mathbf{e}$. The initial distribution is given by $x_{i,P}(0) = e_i x_P(0)$ where $x_P(0)$ is according to (3.28). Using $\mathbf{e}$ for seeding $\mathbf{x}(0)$ gives a first unrefined, but computationally inexpensive approximation of $\mathbf{x}^*$.

Environment step after generation step. In the previous sections, we discussed an evolutionary system residing in an environment defined by transition matrix E. Before every generation step of the system, we allowed the environment to change to a new state. We can turn this order around, and make a step in the state space of the environment after computing a new generation, with

$$T'_{sto} = \mathbf{ET} = \begin{pmatrix} E_{1,1}T_1 & \cdots & E_{1,n}T_n \\ \vdots & \ddots & \vdots \\ E_{n,1}T_1 & \cdots & E_{n,n}T_n \end{pmatrix} . \tag{5.18}$$

Note that this matrix differs from the resulting transition matrix of (5.10) in the transposed positions of T_i blocks.

We now would like to find eigenvector $\mathbf{y}^*$, with corresponding eigenvalue 1, of $T'_{sto} = \mathbf{ET}$, based on our knowledge that $\mathbf{x}^*$ is the eigenvector, also with corresponding eigenvalue 1, of $T_{sto} = \mathbf{TE}$. In the definition of eigenvector $\mathbf{x}^*$, i.e., $\mathbf{x}^* = \mathbf{TEx}^*$, we can multiply both the left and right hand side with $\mathbf{E}$, such that $\mathbf{Ex}^* = \mathbf{ETEx}^*$ also holds. Combining this with the definition of eigenvector $\mathbf{y}^*$ where $\mathbf{y}^* = \mathbf{ETy}^*$, gives $\mathbf{y}^* = \mathbf{Ex}^*$. This gives us the eigenvector of T'_{sto} as a simple transformation, or a stochastic environment step according to E, of the eigenvector of T_{sto}.

Throughout the rest of this chapter we work with transition matrix $T_{sto} = \mathbf{TE}$ and its eigenvector with corresponding eigenvalue 1 to discuss the limit distribution of the evolutionary system in the dynamic environment. If we would use $\mathbf{ET}$ as transition matrix, the resulting eigenvector would depict the distribution over the states of the algorithm after an environment step had been taken. On the other hand, by studying the eigenvector of $\mathbf{TE}$, we allow the algorithm to step one more generation after the environment has stochastically changed. Comparing both distributions would allow us to study the performance of one generation of the algorithm after the environment has changed.

5.3.5 Expected performance

Let $\mathbf{x}^*$ denote the eigenvector with corresponding eigenvalue 1 of matrix product $\mathbf{TE}$, for a given stochastic dynamic environment and evolutionary system with its parameters. Element $x^*_{(f_i,P)}$ of $\mathbf{x}^*$ denotes the probability of ending up in state (f_i, P) of the system. We want to construct a measure of performance to discuss the evolutionary system's expected performance in the dynamic environment. In order to do so, we give the expected mean fitness of a population, where any of the fitness functions of the dynamic environment can be active, as given by the fixed point distribution $\mathbf{e}$ of matrix E. This allows us to compare the expected mean fitness for evolutionary systems with different parameters, and the influence of these parameters on the performance of the optimizing algorithm.

Let $\overline{f_i}(P)$ denote the average fitness of population P according to fitness function f_i. The weighted mean $\overline{f}$ of population fitnesses in states (f_i, P), according to fixed point distribution $\mathbf{x}^*$, is given by

$$\overline{f} = \sum_{(f_i,P)\in F\times\pi} x^*_{(f_i,P)}\overline{f_i}(P). \tag{5.19}$$

We use $\overline{f}$ as a measure of performance, denoting the expected mean fitness of a population in the dynamic environment, in the fixed point of $\mathbf{TE}$.

5.4 Alternating the Deleterious Allele

In order to show how the models and methods can be utilized in practical applications to study the performance of distinct algorithms and their parameters in dynamic optimization, we discuss a dynamic problem that can be expressed similarly as a deterministically and stochastically dynamic fitness environment. We then study the limit behavior of the evolutionary systems for these environments and review the influence of the model's parameters on the expected performance toward the limit of a haploid and diploid evolutionary system tracking the dynamic problem. In similarity with the observations of Chapter 4 we compare the finite population models with their corresponding models with infinitely large populations.

We construct a small problem in a dynamic environment whose search space has a single locus and two alleles. In terms of bit strings, this implies that the algorithms have a bit string of length 1, or 1 bit, as their phenotype. We use $n = 2$ different fitness functions for our dynamic problem. Alternately, we let 0 and 1 be the deleterious allele for g generations, on average. Dependent on whether the stochastic or deterministic approach is used when modeling the dynamic fitness environment, this number respectively denotes the expected or exact number of consecutive generations that either fitness function is alternately used for selection.

Let f_1 and f_2, with $f_1(0) = 1 - L, f_1(1) = 1, f_2(0) = 1$ and $f_2(1) = 1 - L$, be the alternating fitness functions, with L denoting a measure of selection pressure, with $0 \leq L \leq 1$. In f_1, allele 1 is advantageous, and in f_2 it is the deleterious allele. As L is increased, the selection pressure is also higher. Let T_1 and T_2 denote the transition matrices for one generation of an evolutionary algorithm whose selection is proportional to f_1 and f_2, respectively.

5.4.1 Deterministic environment

In the deterministically dynamic environment, we let fitness function f_1 be active for g generations, after which f_2 governs the selection in the evolutionary system for g consecutive generations. This cycle is then repeated indefinitely and defines the deterministic model of the environment. Matrix

$$T_{det} = T_2^g T_1^g \tag{5.20}$$

represents the transition probabilities of the algorithm for $g_{tot} = 2g$ consecutive generations, starting with the first generation using fitness function f_1. Since the model of the environment is deterministic, each of the fitness functions governs the selection process for exactly g generations at a time. The unique eigenvector of T_{det}, with corresponding eigenvalue 1, of this transition matrix is used as a basis for computing the algorithm's performance for this problem.

5.4.2 Stochastic environment

Analogous to the deterministic environment we construct a stochastic environment that, on average, behaves similarly. In order to accomplish this, we construct the environment as a Markov model defined over the fitness functions, which are states of the model, with transition probability matrix according to

$$E = \begin{pmatrix} 1-\alpha & \alpha \\ \alpha & 1-\alpha \end{pmatrix}. \tag{5.21}$$

Parameter α denotes the probability of switching to the other fitness function in one generation. For $0 < \alpha < 1$, matrix E is irreducible and aperiodic, and has a unique steady state distribution $\mathbf{e} = (1/2, 1/2)^\top$.

The environment remains in a state, or static fitness function for a stochastic number of generations. The distribution over these lengths, or number of generations, is geometric. The probability that the environment remains static for k consecutive generations – before switching fitness functions – is given by $(1-\alpha)^{k-1}\alpha$. The mean number of generations that the model stays with the same fitness function is $1/\alpha$. The standard deviation of the geometric distribution about this mean equals $\frac{\sqrt{1-\alpha}}{\alpha}$.

If $\alpha = 1/g$ then the expected number of generations between changing fitness functions in the stochastic environment is also g. We refer to α as the rate or frequency of alternation.

Block matrix T_{sto}, with

$$\begin{aligned} T_{sto} &= \mathbf{TE} \\ &= \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix} \begin{pmatrix} (1-\alpha)I & \alpha I \\ \alpha I & (1-\alpha)I \end{pmatrix} \\ &= \begin{pmatrix} (1-\alpha)T_1 & \alpha T_1 \\ \alpha T_2 & (1-\alpha)T_2 \end{pmatrix} \end{aligned} \tag{5.22}$$

gives the Markov model implementation of the finite population model in the stochastic fitness environment as in (5.10), according to fitness functions f_1 and f_2 and the environment's Markov matrix E from (5.21).

As a default setting in the observations, we assume $\alpha = 0.1$, or $g = 10$, i.e., the expected number of generations that a static fitness function is active is 10. The deviation about this mean is 0 for the deterministic environment, obviously, and $3\sqrt{10}$ for the stochastic environment. For our example environment, this default frequency of changes offers the finite population to discover and exploit advantageous alleles. Later, we vary the rate of changes to observe the impact of this parameter on the performance of the finite population.

5.5 Results

In this section, we study the limit behavior of the evolutionary system in deterministic and stochastic dynamic environments with an alternating deleterious allele, as defined in the previous section. We discuss the influence of the parameters of

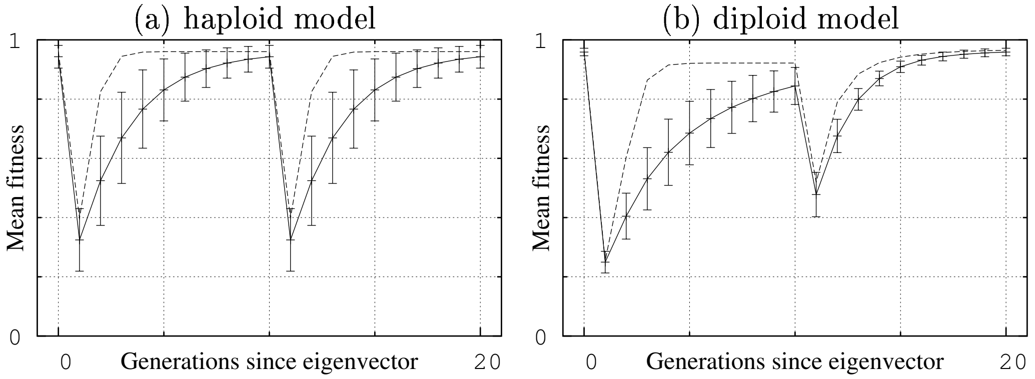


Figure 5.1: Expected performance of the (a) haploid and (b) diploid algorithm in the deterministically dynamic environment with $r = 10, \mu = 0.04, L = 0.9, \alpha = 0.1, h = 0$. Mean fitness and standard deviation of the distribution are given for each of the 20 generations of the model, where the first generation is derived from the eigenvector. The graphs in dashed line style represent the behavior of the corresponding infinitely large population model, as a reference.

the model – ploidy and coefficient of dominance, population size, mutation rate, rate of alternation and selective pressure – on the performance of the algorithm in its deterministically and stochastically dynamic fitness environment.

First, we study some of these parameters in the deterministic environment, as the specific interpretation of its limit behavior allows us to better comprehend how a change in a given parameter affects the limit behavior. We then compare those results in the deterministic environment with their corresponding predictions in the stochastically dynamic environment. Since the models with deterministically and stochastically dynamic environments are similar, predictions for the evolutionary system with a deterministic environment generally also apply to the corresponding model with a stochastic environment.

5.5.1 Interpretation of the limit

First, consider the graphs in Figure 5.1. These figures depict the typical limit behavior of specific instantiations of the haploid and diploid model in the deterministically dynamic environment. The parameters for both models are set to population size $r = 10$, mutation rate $\mu = 0.04$, selective pressure $L = 0.9$ and the rate of alternation is set to $\alpha = 0.1$, i.e., there is a period of 10 generations between switching from one fitness function to the other. For the diploid model, allele 0 is dominant with dominance coefficient $h = 0$, i.e., allele 0 is dominant over allele 1 with probability 1.

For each of the generations u, with $0 \leq u \leq 2g$, the expected mean fitness, or expected fitness $\overline{f_{(u)}}$ – and the standard deviation about this expectation – of the distribution over the populations at u generations from the eigenvector has been plotted. At generation 0, the eigenvector of the model itself is represented.

In the first 10 generations of both figures, selection in the model is governed by f_1, where allele 0 is the deleterious allele and 1 is the advantageous allele. In the latter 10 generations, the second fitness function f_2 governs the selection process of the model, with allele 0 being the advantageous allele. At 20 generations from the eigenvector, we have returned to the eigenvector of the model, by definition, and the system repeats this behavior indefinitely in the long run of the system. The 20 consecutive distributions thus represent the limit behavior, as each of the distributions over the populations for each of the generations is unrolled from the eigenvector as in (5.4).

As fitness functions switch, i.e., at generations $0, 10, 20, ...$ since the eigenvector, the expected mean fitness of populations in the distribution is reduced to $(2 - L) - \overline{f}$, and the selection and reproduction in the model proceeds according to the new fitness function. Since the fitness function changes every 10 generations, the algorithm is repeatedly *restarted*. The repetitive behavior in the model is similar to the transient behavior of an evolutionary system with a static fitness function, even in the limit. Indeed, as the fitness functions switch, and the advantageous and deleterious alleles switch, the result of optimizing for the previous fitness function is lost, and the algorithm has to start over again, searching for the advantageous allele that was the deleterious allele before the switch. As the periods in between fitness function alternations become longer, the algorithms are given more time to collect advantageous alleles in their populations. As the populations are given more time to evolve for one static fitness function in between alternations, they can build up larger proportions of advantageous alleles. But, as their mean fitness becomes higher, the mean fitness also drops lower when the fitness function is eventually switched. In the limit behavior of the model, the behavior has found a balance between such drops and growths in proportions of advantageous alleles.

In the haploid model, the populations react similarly to either switch of the fitness functions, as the haploid algorithm behaves the same as genotypes 0 and 1 are treated similarly. This can be observed in the figures, as both halves in the graph representing the limit behavior of the haploid model are the same. In contrast, the diploid algorithm reacts differently when the deleterious allele is changed from 0 to 1 as compared with the reverse switch. This is due to the fact that the diploid algorithm isn't symmetrical in its behavior because of the asymmetric dominance operator, as was already noted in Section 4.3. As in the model with static selection, the diploid model is able to perform better when the advantageous allele is also the dominant allele in the dominance scheme, thus reacting differently when the fitness functions are switched. This behavior can be observed in Figure 5.1(b), where the diploid evolutionary system better recovers from the fitness function switch when the dominant allele becomes advantageous, as compared to the dominant allele becoming deleterious.

Figure 5.2 shows the influence of a varying coefficient of dominance. The heterozygous genotype $\{0, 1\}$ has phenotype 0 with probability h, and phenotype 1

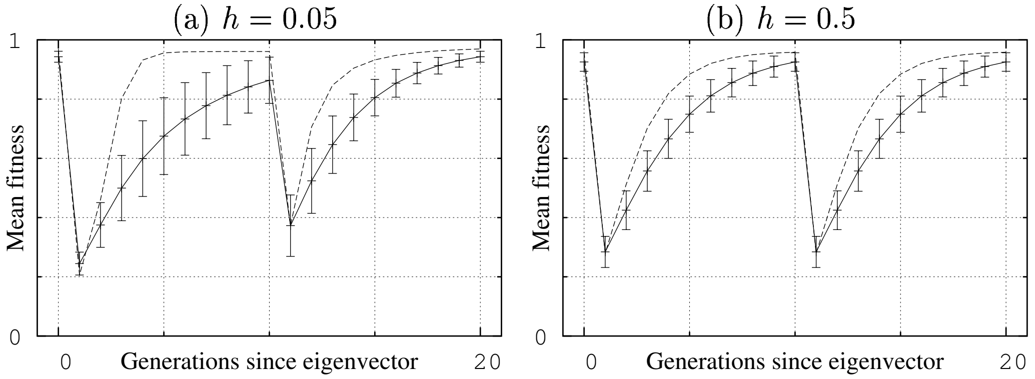


Figure 5.2: Expected performance of the diploid algorithm in the deterministically dynamic environment with $r = 10, \mu = 0.04, L = 0.9, \alpha = 0.1$ with varying coefficient of dominance $h = 0.05$ and $h = 0.5$. Mean fitness and standard deviation of the distribution are given for each of the 20 generations since the eigenvector of the model.

with probability $1-h$. It can be observed that the diploid algorithm gathers more fitness in the dynamic setup as the dominance degree goes to 0.5. In the dynamic setup, a diploid population is able to maintain a higher diversity of individuals, which allows the population to adapt to the changing fitness environment. As the dominance coefficient is increased, the variance of fitness in the population is higher. The asymmetry in the graphs that arose as diploidy was introduced, disappears again as $h = 0.5$.

5.5.2 Frequency of alternation

Figure 5.3 gives the expected performance of the algorithms in their respective environments for varying periods in between fitness function switches. It can be observed that as the frequency of alternations is decreased, i.e., as the period in between alternations is increased, the algorithm can perform better. The fitness gained during the optimization in between alternations outweigh the loss of fitness when switching between fitness functions. Note that this is true except for extremely high frequencies of alternation, which we discuss later on. For sufficiently long periods, the finite model also becomes more similar to the infinitely large population, i.e. it performs better. In the transient behavior of a finite model, it was noted that the finite model lags behind after the infinite population model. The effect that a finite model loses some performance over the infinite population model can also be observed in the model with a dynamic fitness function, as the long run behavior of the evolutionary system in a dynamically changing fitness environment behaves as the initial transient behavior of an evolutionary system in a static fitness function. As the periods between fitness function switches become longer, the distribution over the populations just before a fitness switch is closer to the limit distribution of the system in its static environment.

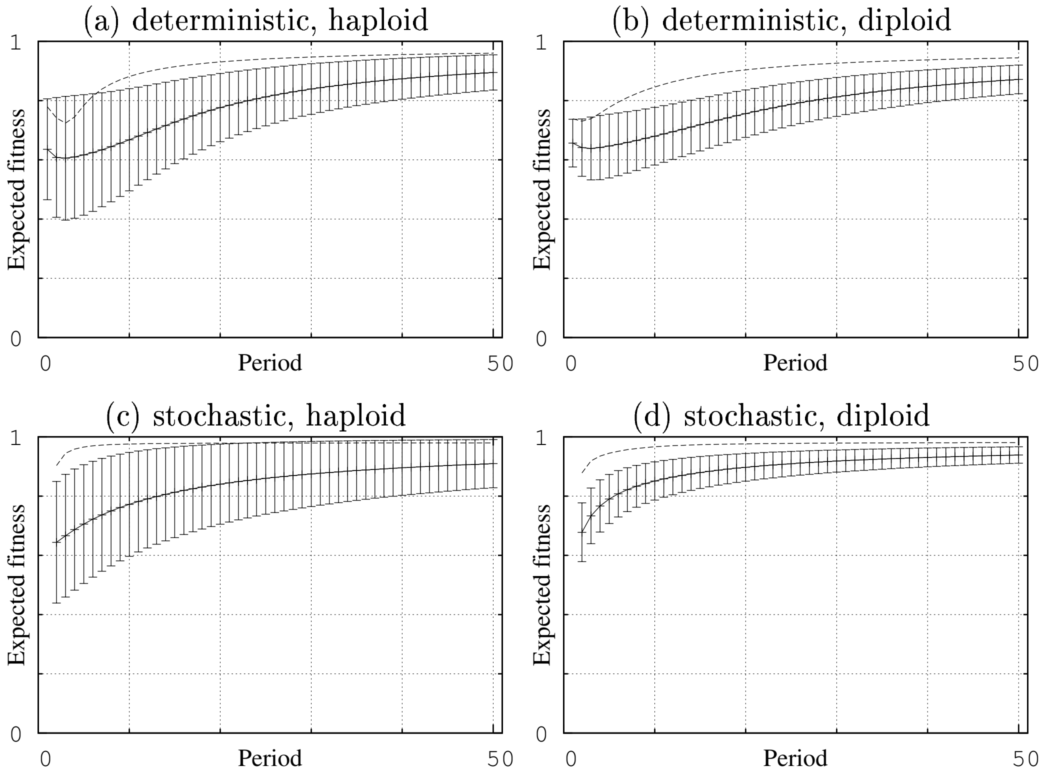


Figure 5.3: Expected long run performance and standard deviation of the system for different mean periods between fitness function alternations with $r = 10, \mu = 0.02, L = 0.9, h = 0$. The graph in dashed line style represents the limit behavior of the corresponding infinitely large model.

In the extreme case that alternations rapidly proceed each other, some odd effects of these extreme frequencies can be observed. In the deterministic environments, if alternations follow each one another very fast, the algorithm doesn't evolve good proportions of advantageous alleles in the short time between alternations, and consequently, the proportions do not drop to extreme low numbers at a switch. For relatively fast environments, the loss of good proportions at a fitness function switch may outweigh the gains from evolving good proportions of advantageous alleles in between fitness function alternations. This effect can be observed in Figures 5.3(a, b) as there is a small drop in expected performance for periods of length 2 to about 5, in comparison with periods of length 1.

In comparison, the stochastic model does not show such behavior as the lengths of periods are smoothed out because of a high variance about the mean period. Note that the stochastic model with period $\alpha = 1$ results in an environment with a transition probability matrix E according to

$$E = \begin{pmatrix} 1-\alpha & \alpha \\ \alpha & 1-\alpha \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

which is aperiodic. If combined with the transition probability matrices of the evolutionary system, the model would also become aperiodic and is thus not considered as a viable environment for our model. The expectations of the evolutionary system in the stochastic environment with $\alpha = 1$ is consequently absent from Figures 5.3(c, d).

5.5.3 Population size

Figure 5.4 shows that the evolutionary system performs better when the population size is increased. Indeed, the effect of sampling errors, and thus genetic drift, become less prominent, and the behavior becomes more similar to the behavior in the infinite population model. As the population size of the system decreases, genetic drift becomes more prominent. In the limit case that $r = 1$, the expected performance of the system becomes $1 - L/2$.

In practical applications of genetic algorithms, larger populations require more computational effort for making the step to the next generation. This has its effect on the model when dynamic optimization problems are studied. An implementation of a genetic algorithm usually requires $O(n)$ computations for a population of size n to advance one generation. Suppose – for an exemplary online optimization problem – that a population of 10 individuals experiences a change in the fitness function every 20 generations. For the same problem, a genetic algorithm with a population of 20 individuals would encounter a change in the online environment at every 10th generation. An implementation of a genetic algorithm processes generations slower as their population is increased. In order to take care of this difference in the model, one could choose to speed up the dynamic environment as the population becomes larger. We ignore this as our primary goal is to build models of biological populations, where no computational effort has to be taken into account. Indeed, in biological populations, selection and reproduction proceeds inherently parallel. Consequently, a generation in a biological population of 10 or 20 individuals requires the same amount of time with respect to the dynamic environment.

5.5.4 Mutation rate

Figure 5.5 shows the influence of varying the mutation rate on the performance of the system for the dynamic problem. In similarity with static fitness functions, the effects of genetic drift – i.e., the difference between the infinitely large population model and the finite population model because of sampling errors – are more prominent with smaller mutation rates. The reasoning is the same as before. As the mutation rate decreases, the system has a higher probability of getting fixated in the population whose individuals all have the deleterious allele, since the rate of mutants invading the population is slower than the rate of fixation of alleles. As a result, the population is stuck at a population with one of either alleles

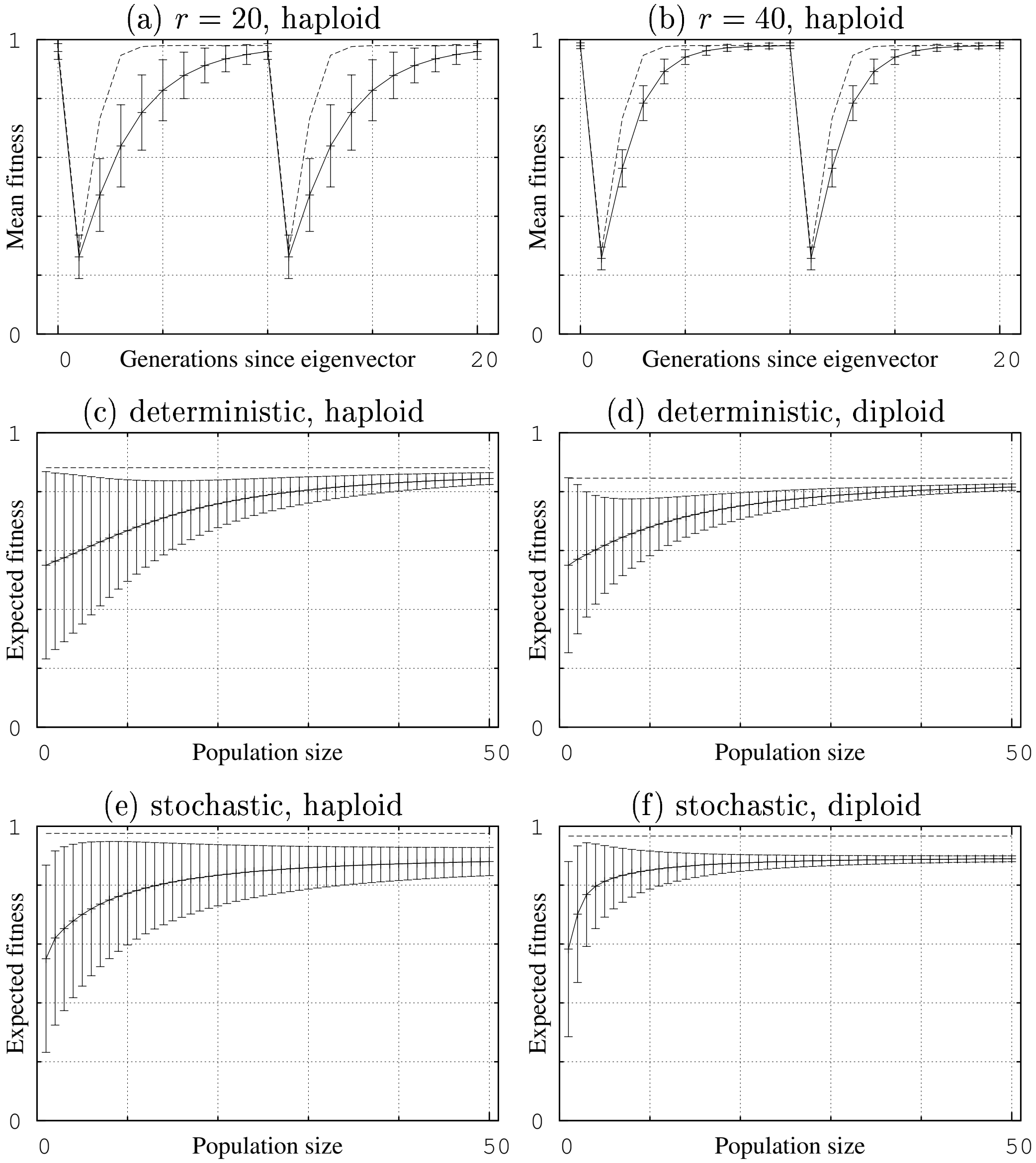


Figure 5.4: Expected performance for different population sizes with $\mu = 0.02, L = 0.9, \alpha = 0.1, h = 0$. (a-b) Mean fitness and standard deviation of the distribution at each of the 20 generations since the eigenvector of the model, for $r = 20, 40$ in the deterministic environment. (c-f) Expected performance and standard deviation of the exact limit behavior of the system for population sizes between 1 and 50.

fixated, and can barely escape from this situation, independent of the selective pressure that is active on the system.

Because of the alternation of the fitness functions, the repetitive behavior in the limit is similar to the initial transient behavior of the evolutionary system with a fixed fitness function. As such, in the limit, the choice of the mutation rate is of importance to react optimally to changes in the environment. In the

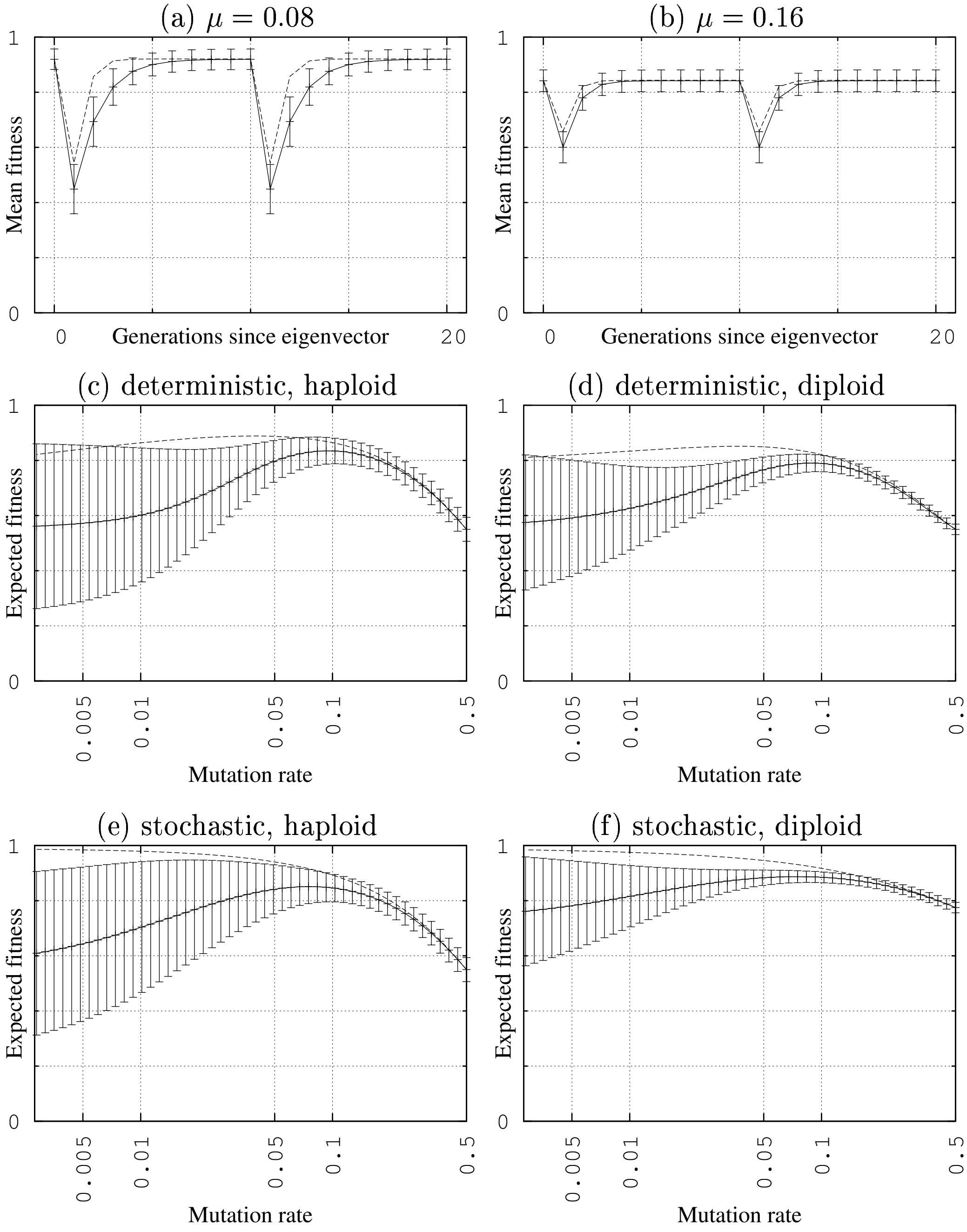


Figure 5.5: Expected performance of the haploid algorithm in the (a-d) deterministically and (e-f) stochastically dynamic environment for different mutation rates with $r = 10, L = 0.9, \alpha = 0.1$. The graphs in dashed line style represent the behavior of the corresponding infinitely large population model, as a reference. (a-b) Mean fitness and standard deviation of the distribution at each of the 20 generations since the eigenvector of the model for $\mu = 0.08, 0.16$. (c-f) Expected performance and standard deviation of the exact limit behavior of the system for mutation rates between 0.003 and 0.5.

limit behavior of fixed fitness functions, a smaller mutation rate gives better performance for the system, where a relatively higher mutation rate is better suited for the initial transient behavior of the optimization algorithm, if fixed fitness functions are used.

If the mutation rate is relatively small, the algorithm has no means of changing the population's focus from one allele to the other if the deleterious allele changes. On the other hand, if the mutation rate is sufficiently high, too many deleterious mutations enter the populations even before the environment has changed. Note that this is also visible in the standard deviations of the mean fitness. If the mutation rate is sufficiently small, the standard deviation is very large, indicating that populations get stuck with high proportions of either deleterious or advantageous alleles. As the fitness function changes the deleterious bit over time, the standard deviation thus remains high if the mutation rate is low. On the other hand, if the mutation rate is fairly high, the standard deviation of the mean fitness is fairly low, indicating that the algorithm mainly generates populations with the same compositions, i.e., populations that are highly diverse, since mutation mainly governs the optimization process. This may sound contradictory at first, as high standard deviations are expected for highly diverse populations. However, we measure the standard deviation in between populations according to the distribution over the populations, not the "intra" population standard deviation. The low standard deviation arises from the fact that only one type of population (the most diverse population) is mainly present in the distribution over the populations.

The mutation rate must thus be balanced according to the speed of changes in the dynamic environment for the best performance of the algorithm.

As an example, for the haploid system with $r = 10$, $L = 0.9$ and $\underline{\alpha} = 0.1$, a bitwise mutation rate of $\mu \approx 0.095$ gives the best performance, with $\overline{f} \approx 0.835$. The expected performance as the mutation rate approaches $\mu = 0$ or $\mu = 0.5$ is $\frac{2-L}{2} = 0.55$. Indeed, in the case that $\mu = 0$, the population would become fixated in one of the alleles. Each individual then receives fitnesses 1 and $1 - L$ for an equal amount of time. At generations when the fixated allele of the individual is advantageous or deleterious it receives fitness 1 or $1 - L$, respectively. The expected fitness as $\mu = 0$ is thus the average of 1 and $1 - L$, i.e., $\frac{2-L}{2}$. Similarly, as the mutation rate equals $\mu = 0.5$, the expected proportion of either allele, at any generation is 0.5. The expected fitness of such a population is also the average of 1 and $1 - L$, $\frac{2-L}{2}$.

The optimal mutation rate for the infinite population model is lower, as compared to that of a finite population. As populations get smaller, higher mutation rates are required to perform optimally, to counter for the more prominent effects of random sampling errors. If the frequency of alternation increases, higher mutation rates perform more optimally as with lower frequencies, as more exploration is required in these rapidly changing environments.

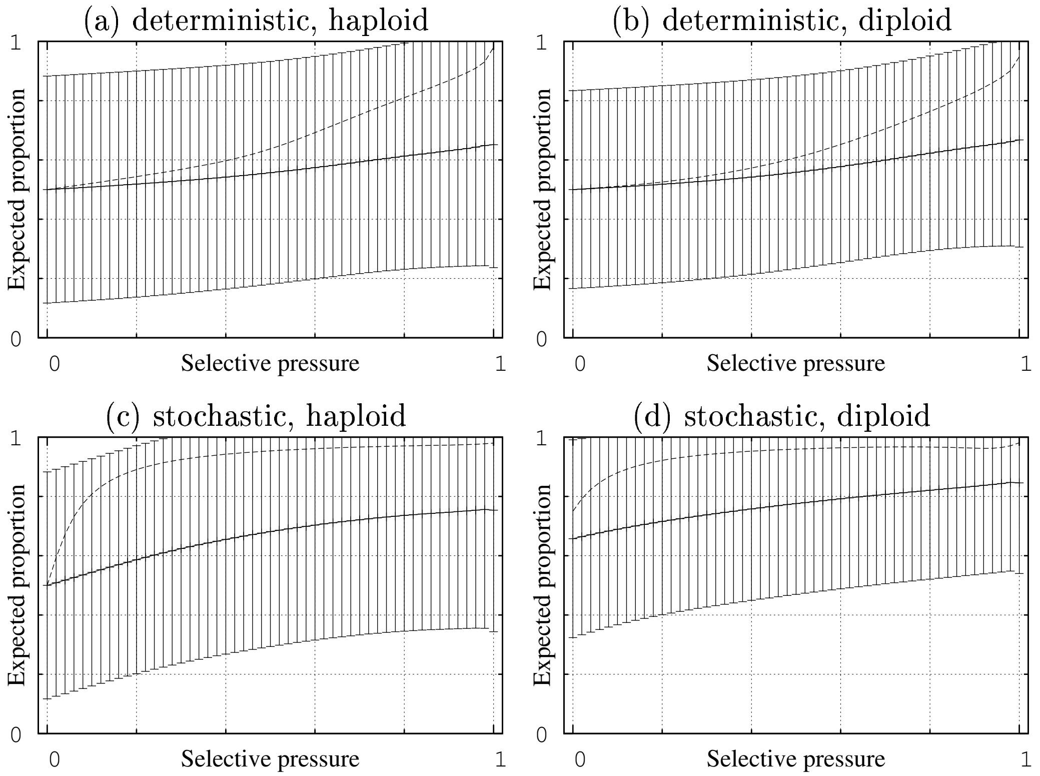


Figure 5.6: Expected performance and standard deviation of the algorithm in its dynamic environment for varying selective pressure with $r = 10, \mu = 0.02, \alpha = 0.1, h = 0$.

5.5.5 Selective pressure

Figure 5.6 gives the expected proportion of optimally fit individuals in the long run for different settings of selection pressure L. We have chosen not to compute and show the expected mean fitness, as it would give a distorted comparison of the performance of the algorithms as the fitness functions for all data points are different. Note that these two measures of performance – proportion of optimally fit individuals and mean fitness – can be converted into each other as the search space constitutes only two types of individuals.

As the selection pressure is lower, or as the fitness of deleterious individuals is closer to the fitness of advantageous individuals, the algorithms tend to perform worse in any of the proposed environments. If more selective pressure is present, the evolutionary system is more focused on gathering higher proportions of the advantageous allele.

For static fitness functions, we have observed that the behavior in the finite population model deviates more from the infinite model for lower selective pressures. This is no longer true in the case of dynamic fitness environments. For small selective pressures, the behavior of the finite model is somewhat similar to the infinite model. For large selective pressures, the deviation of the finite model with respect to the infinite model is more prominent. This difference is due to the fact that in the model with dynamically changing fitness, the behavior is governed by short periods of transient behavior. Finite populations require more time to gather good proportions of fit alleles, or to escape from a state of fixation in comparison to the infinite model. The transient behavior of an infinitely large population performs best under high selective pressure. The infinite model can generously benefit from the difference in fitnesses between the advantageous and deleterious alleles. Indeed, if only a small proportion of the fitter allele is present in the infinite population after a switch of fitness functions, then this proportion can easily take over the population. The finite model is plagued by the effects of fixation. We can observe that the effect of fixation is stronger than the selective force which allows fitter populations to invade the population. As the fitness environment changes over time, the finite model does not have enough generations to gather high proportions of the advantageous allele, independent of the selective pressure, where the infinite model performs best under high selective pressure.

Summary

We have constructed Markov models of evolutionary systems with finite populations in dynamic environments. These environments assume continuous deterministic or stochastic alternation of a set of fixed fitness functions. The models allow us to study the influence of different parameter settings on the expected performance of the system in the long term. The model also offers a possibility to compare the influences on the behavior for both static and dynamic fitness environments. An exemplary dynamic environment, where an allele alternates between being advantageous and deleterious has been studied.

At each switch of the fitness function, the population is required to re-explore the genotype space from scratch. Finite populations become fixated in the advantageous allele before a switch which causes the allele to become deleterious. Since the population is fixated in the deleterious allele at each switch, a high rate of exploration, and thus of mutation, is required. This is in contrast with static fitness environments, where a sufficiently low rate of mutation was required to become fixated in the advantageous allele in the long term. The pressure to escape from fixated populations is higher with smaller populations. Note that the difference between finite and infinite populations is thus more prominent in dynamic environments. As environments change faster, higher rates of mutation are required to perform best.

Only a slight advantage for diploid populations has been observed, for slow rates of environmental change. For the 1 bit problem considered in this chapter, the differences between the expected performance of haploid and diploid populations is not statistically significant.

Note that the model with a deterministic environment only details changing fitness functions that are assumed to be cyclic, since we need it as a prerequisite to construct our Markov models and to discuss the limit behavior of the model. Other functions that determine the dynamics of a fitness environment may not necessarily be cyclic or periodic, but might be of interest to the reader. Generally, we cannot predict how the population behaves in an arbitrary changing environment, which is defined as a function over time. Only in the case that the environment's state converges to a static environment in the long term, we can say something about its limit behavior. Indeed, in the long run, the model in such environments behaves in similarity to the limit behavior of the population in the static environment that constitutes the environment's state of convergence in the long run.

Chapter 6

Frequency Dependent Fitness Environments

For the analysis of the dynamics of game playing populations, it is common practice to assume infinitely large populations. Infinite models yield predictions of fixed points and their stability properties. However, these models cannot demonstrate the influence of genetic drift, caused by stochastic sampling in small populations. Instead, we propose Markov models of finite populations for the analysis of genetic drift in games. With these exact models, we can study the stability of evolutionary stable strategies, and measure the influence of genetic drift in the long run. We show that genetic drift can introduce significant differences in the expectations of long term behavior.

We construct models of finite populations whose fitness functions are based on evolutionary games. We study the behavior of these models in a Neutral game, the Hawk-Dove game, the Prisoners' Dilemma, a Cooperation game with Risk Dominance and Payoff Dominance, and in the Rock-Paper-Scissors game.

Parts of this chapter are derived from A.M.L Liekens, H.M.M. ten Eikelder, P.A.J. Hilbers, *Predicting Genetic Drift in* 2×2 *Games* (2004).

6.1 Introduction

An increasing number of authors (Schaffer 1988; Foster & Young 1990; Fogel & Fogel 1995; Fogel, Fogel, & Andrews 1997; Fogel, Andrews, & Fogel 1998; Ficici & Pollack 2000a; Miękisz 2005a; Nowak & Sigmund 2004; Nowak, Sasaki, Taylor, & Fudenberg 2004; Neill 2004; Taylor, Fudenberg, Sasaki, & Nowak 2004) have stressed the importance of including finite population size and mutation rates in studies of how populations react in the setting of evolutionary games.

It has been observed that stochastic sampling effects of finite population size may seriously affect the predictions made by Maynard-Smith's (1982) concept of an evolutionary stable strategy (ESS). A strategy is an ESS if a population playing according to this strategy cannot be invaded by a mutant strategy. This concept of the ESS only holds under the assumptions of infinite population size and no variation. Riley (1997) and Vickery (1987) show that a finite population that constitutes of individuals playing the ESS can be invaded. There is indeed a small probability that the less optimal mutant strategy in a finite population takes over the population through an "unlucky" sample that includes this mutant and none of the more prominent strategies in the population. As a result of this "unlucky" sample there exists a small non-zero probability that the population can be invaded. Since this small probability is the result of stochastic sampling, and is consequently non-existent in a deterministic infinite population model, the concept of an ESS thus falls apart once a finite population size is adopted.

A couple of interpretations of the ESS for finite population models have resulted from these observations. Foster & Young (1990) have proposed stochastically stable equilibriums, which is the equilibrium that is preferred by the system as stochastic noise goes to 0. Schaffer (1988) computes the deviation from the ESS as a function of population size, and concludes that the correction can be neglected in populations that are finite but large enough. Neill (2004) observes that there are games in which a strategy that meets the ESS criteria is not evolutionarily stable, no matter how big the population, and proposes criteria for a large population ESS.

While studying a specific game, Miekisz (2005a) concludes that the population size and mutation rate have an important effect on the resulting behavior of the population, such that the specific settings of these parameters must be included in the predictions of equilibrium behavior. Since this observation is the focus of this text, we go deeper in understanding the influence of finite populations in such games. The specific characteristics of the equilibrium behavior of a system with its parameters cannot always clearly be determined without computing the exact equilibrium behavior of the finite population model. Studying the equilibrium behavior solely by determining whether the behavior converges to certain characteristics for extreme values of the mutation rate and population size is not sufficient to predict the equilibrium behavior of the system with its specific parameters. These extreme parameter settings have been studied by evolutionary game theorists and in similarity with the previous sections we adopt our methodology to give predictions for the parameters in between these extremes.

It is thus important to determine the equilibrium behavior of a population in a game for its specific settings of population size and mutation rate. In this chapter, we adopt the mathematical modeling approach as in the previous chapters to study the influence of mutation rates and the population size on the behavior of finite populations for a selection of well-known symmetric 2×2 and 2×3 games. We study the limit behavior of finite populations in a Neutral game, the Hawk-Dove game and the Prisoners' Dilemma. We also study a 2×2 coordination game that incorporates payoff and risk dominance. As a last game, we study the influence of finite population size on the evolutionary dynamics of a well-known 2×3 game, the Rock-Paper-Scissors game. All models assume a limited number of strategies (2 or 3) that can be employed. As such, the strategies can be represented as an atomic genotype with 2 or 3 alleles representing either strategy, which simplifies our analysis. Extended overviews of these games in an infinite population context can be found in (Weibull 1995; Hofbauer & Sigmund 1998).

6.2 Frequency Dependent Fitness

6.2.1 Payoff matrices

All games are represented by a set of strategies Ω and a square payoff matrix A. Each entry $A_{i,j}$ in this payoff matrix gives the payoff value for an individual adopting strategy i when confronted with an individual playing strategy j. We assume that all payoffs in matrix A are nonnegative.

6.2.2 Fitness

Let the fitness of an individual $i \in \Omega$ in its population P be dependent on the payoff matrix A, and the frequencies of the strategies in the population, with

$$f(i, P, \mathcal{E}) = \sum_{j \in \Omega} A_{i,j} p_j. \tag{6.1}$$

We assume there are no further environment variables besides the population P in which the individuals reside in. The fitness denotes the mean payoff received when the individual is matched against all individuals in the population, including itself. As such, the fitness of an individual denotes the expected payoff received when players are randomly chosen as opponents. Note that the fitness used in this chapter is structurally different from the fitness functions in the previous chapters. The fitness of an individual is now dependent on the composition of the current population, instead of relying only on the individual's genome. As evolution is governing the dynamics of the population, the fitness of an individual – i.e., the expected payoff of its strategy – varies over time according to the evolutionary changes in the population. Where the populations in the previous chapters strive towards finding an optimal situation, where their fitness is optimal according to an external fitness function, games may offer a whole different range of dynamics. Populations may evolve to find an equilibrium situation in which the proportion of strategies in the population is evolutionary stable. On the other hand, proportions of strategies may also oscillate around an equilibrium, which is the case in the Rock-Paper-Scissors game. We develop a finite population model of Rock-Paper-Scissors later in this chapter.

6.2.3 Model

We adopt expected payoff as the fitness function in the models of Chapter 3 such that the behavior of individuals playing games in an evolutionary context can be modeled. We assume selection proportional to fitness, and mutation dependent on a rate μ. No crossover is assumed as the genomes, or strategies used in the games are atomic. The results of these models can however be generalized for more complex games and genotypes.

The mean fitness of a population P is given by

$$\overline{f}(P) = \sum_{i\in\Omega} p_i \sum_{j\in\Omega} A_{i,j} p_j. \tag{6.2}$$

The probability of selecting a strategy i from that population is then given by

$$S(i,P) = \frac{p_i \sum_{j\in\Omega} A_{i,j} p_j}{\overline{f}(P)}. \tag{6.3}$$

According to a mutation rate μ, we can determine the probability of producing strategy i from the population. The probability that i occurs in the population of the next generation is given by

$$G(i,P) = (1-(|\Omega|-1)\mu)S(i,P) + \sum_{j\in\Omega, i\neq j} \mu S(j,P). \tag{6.4}$$

For a 2×2 game, i.e., with two players and two strategies 0 and 1, we can write the heuristic function in terms of the proportions of one strategy, by

$$G(0,P) = (1-\mu)S(0,P) + \mu(1-S(0,P))$$

where

$$S(0,P) = \frac{p_0 p_0 A_{0,0} + p_0 p_1 A_{0,1}}{p_0 p_0 A_{0,0} + p_0 p_1 A_{0,1} + p_1 p_0 A_{1,0} + p_1 p_1 A_{1,1}} \tag{6.5}$$

or

$$S(0,P) = \frac{p_0^2 A_{0,0} + p_0(1-p_0)A_{0,1}}{p_0^2 A_{0,0} + p_0(1-p_0)(A_{0,1}+A_{1,0}) + (1-p_0)^2 A_{1,1}} \tag{6.6}$$

denotes the probability that strategy 0 is selected from the population.

6.3 Neutral Game

The first game we consider is without selection, and is adopted for control measurements. We model a Neutral game with two strategies $\Omega = \{0,1\}$ with a payoff matrix A where all payoffs $A_{0,0} = A_{0,1} = A_{1,0} = A_{1,1} = c$ are a constant $c > 0$, i.e., the payoff matrix of the Neutral game is

$$A = \begin{pmatrix} c & c \\ c & c \end{pmatrix}. \tag{6.7}$$

Note that an evolutionary system according to the Neutral game reduces to the model with one locus, two alleles and no selection, as considered in Section 4.2, since the fitness of individuals adopting either strategy 0 or strategy 1 equals $f(i,P) = A_{i,0}p_0 + A_{i,1}p_1 = c(p_0 + p_1) = c$.

In the setting of EGT, no strategy for the Neutral game is an ESS, since all states of the infinite population model are fixed points of the infinite population system. Recall that a strategy is an ESS of the game if, in the infinite population setting, the strategy cannot be invaded by a mutant strategy. A system with an infinite population has stable fixed points representing a population according to the ESS. Since both strategies in the Neutral game always have equal fitness, all (pure or mixed) strategies can be invaded.

If variation is assumed, the fixed point of the infinite model however is always at the mixed strategy that employs 50% of strategy 0, and vice versa. In the finite model of the Neutral game, however, the populations have been observed to drift away from this fixed point, toward higher probabilities of fixation of either of the two strategies, if relatively small population sizes, or a relatively small mutation rate is assumed. The Neutral game is used as a control measurement to see how selective pressure in other games influences these predictions.

6.4 Hawk-Dove

6.4.1 Game

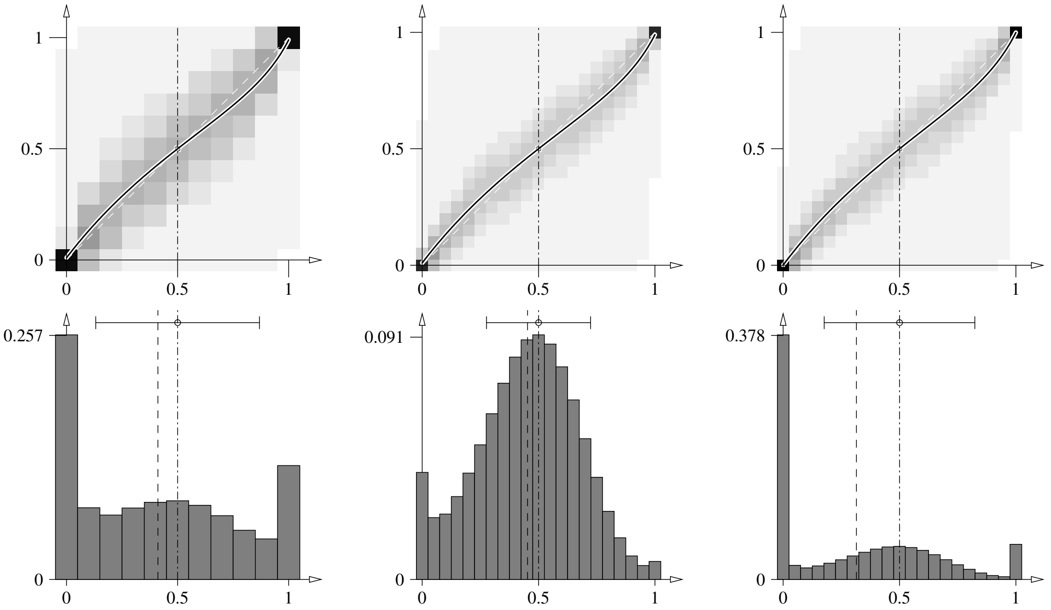
In this game, a bird has the choice between 2 behaviors when a resource needs to be shared with another bird. It can either choose to act as an aggressive Hawk or a pacific Dove. If both players choose the Hawk strategy, they fight and injure each other. If only one of both players chooses Hawk, then this player defeats the pacific strategy of the Dove. If both players play Dove, there is a tie in profit, but the profit is lower than the profit of a Hawk defeating a Dove. The Hawk-Dove game is also known as the snowdrift or chicken game.

The game can be modeled as a game with two strategies $\Omega = \{H, D\}$ (Hawk and Dove), with a payoff matrix A where $A_{H,H} < A_{D,H} < A_{D,D} < A_{H,D}$. Both populations that constitute of either strategy are unstable fixed points of the game if an infinite population without variation is assumed. There also exists a mixed strategy that is an ESS of the system, as the fitness of individuals adopting the Dove and Hawk strategy is equal (Maynard Smith & Price 1973), i.e., when

$$\begin{array}{rrcl} & f(H,P) & = & f(D,P) \\ \Rightarrow & A_{H,H}p_H + A_{H,D}p_D & = & A_{D,H}p_H + A_{D,D}p_D \\ \Rightarrow & (A_{H,H} - A_{D,H})p_H & = & (A_{D,D} - A_{H,D})(1 - p_H) \\ \Rightarrow & (A_{H,H} + A_{D,D} - A_{H,D} - A_{D,H})p_H & = & A_{D,D} - A_{H,D} \end{array} \tag{6.8}$$

The ESS is a population where the proportion of Hawks in the population equals

$$p_H = \frac{A_{D,D} - A_{H,D}}{A_{D,D} - A_{H,D} + A_{H,H} - A_{D,H}}. \tag{6.9}$$



(a) $r = 10, \mu = 0.01$ (b) $r = 20, \mu = 0.01$ (c) $r = 20, \mu = 0.001$

Figure 6.1: Transition matrices and limit or steady state distributions for the Hawk-Dove game for 3 different parameter settings. The horizontal axis represents the proportion of Hawk genomes in the population. The vertical dashed line represents the mean of the distribution, where the dash-dotted line represents the fixed point of the infinite model, as in Figure 4.1. The horizontal error bars represent the expected deviation of the finite population model with respect to the fixed point of its corresponding infinite population model.

Since the absolute value of the denominator of this fraction is larger than the absolute value of the nominator and both have the same sign for the Hawk-Dove game, this mixed strategy always lays in between 0 and 1.

6.4.2 Results

For the exact measurements of the evolutionary dynamics of the Hawk-Dove game, the payoffs game have been chosen with $A_{H,H} = 1 < A_{D,H} = 2 < A_{D,D} = 3 < A_{H,D} = 4$, i.e., we use matrix

$$A = \begin{pmatrix} 1 & 4 \\ 2 & 3 \end{pmatrix}. \tag{6.10}$$

as the payoff matrix for the Hawk-Dove game. The ESS, or stable fixed point of the infinite model with no variation, according to (6.9), for these payoffs lies at $1/2$, with $5/2$ being the fitness of either strategy. Even more, if variation is assumed, the stable fixed point remains at $1/2$ and no other fixed points exist. Indeed, if the population P contains proportions $p_H = p_D = 1/2$ of Hawks and

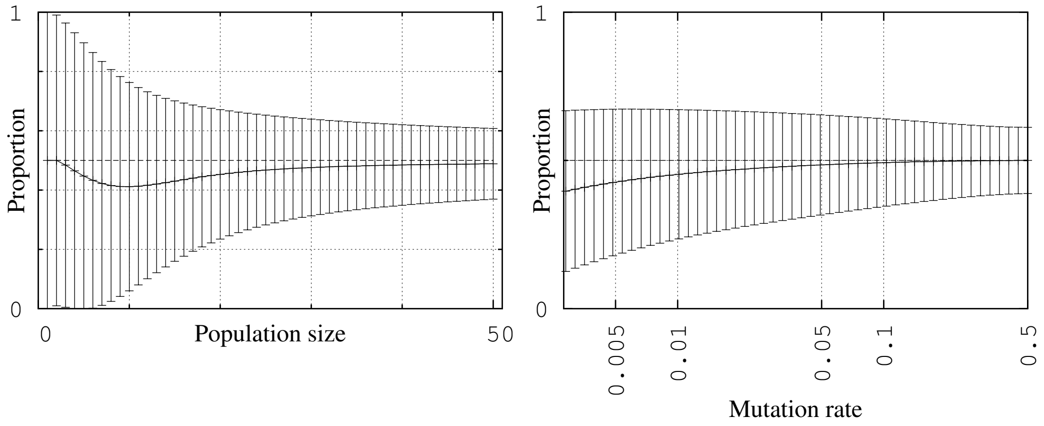


Figure 6.2: Expected proportion of Hawks and standard deviation in the limit distribution of the Hawk-Dove game, for different population sizes with $\mu = 0.01$ (left) and a range of mutation rates with $r = 20$ (right).

Doves, they are both selected with probabilities $S(H, P) = S(D, P) = 1/2$. The probability that a Hawk is generated for the next generation equals

$$G(H, P) = (1 - \mu)S(H, P) + \mu(1 - S(H, P)) = \frac{1 - \mu}{2} + \frac{\mu}{2} = \frac{1}{2} \tag{6.11}$$

which is independent of the value of μ. Note that this observation is specific for the values we have chosen for payoff matrix A. It allows us to study the influence of an increasing mutation rate on the behavior of the finite model without affecting the fixed point of the infinite model.

Consequently, the evolutionary system with this game is similar to the Neutral game, in only having a stable mixed strategy fixed point at 1/2 if variational pressure is assumed. When choosing extremely small or large parameters for population size and mutation rate, the predictions are also similar to those of the Neutral game. Relatively small population sizes, or relatively small mutation rates lead to higher influence of drift on the expectations. Under those parameters, the effects of random sampling, and thus the forces of genetic drift are much stronger, than those of selection according to the payoffs in the game. In the Hawk-Dove game however, selection is asymmetric for either strategy. This allows a finite population to wander away from the infinite model's projected "stable" fixed point, which on its turn may result in drift of the expectations as compared with the infinite model. We can study this effect, and the balance between selective and variational pressure, by examining the differences in expected behavior.

Figure 6.1 represents the transition probability matrices and steady state distributions of the finite population model for the Hawk-Dove game, for three parameter settings of the system. Figures 6.1(a) and (b) show how the system balances between the selection around the fixed point of the infinite model on one hand, and the influence of genetic drift which forces the system to either extreme of its state space on the other hand. Figure 6.1(c) shows how genetic drift

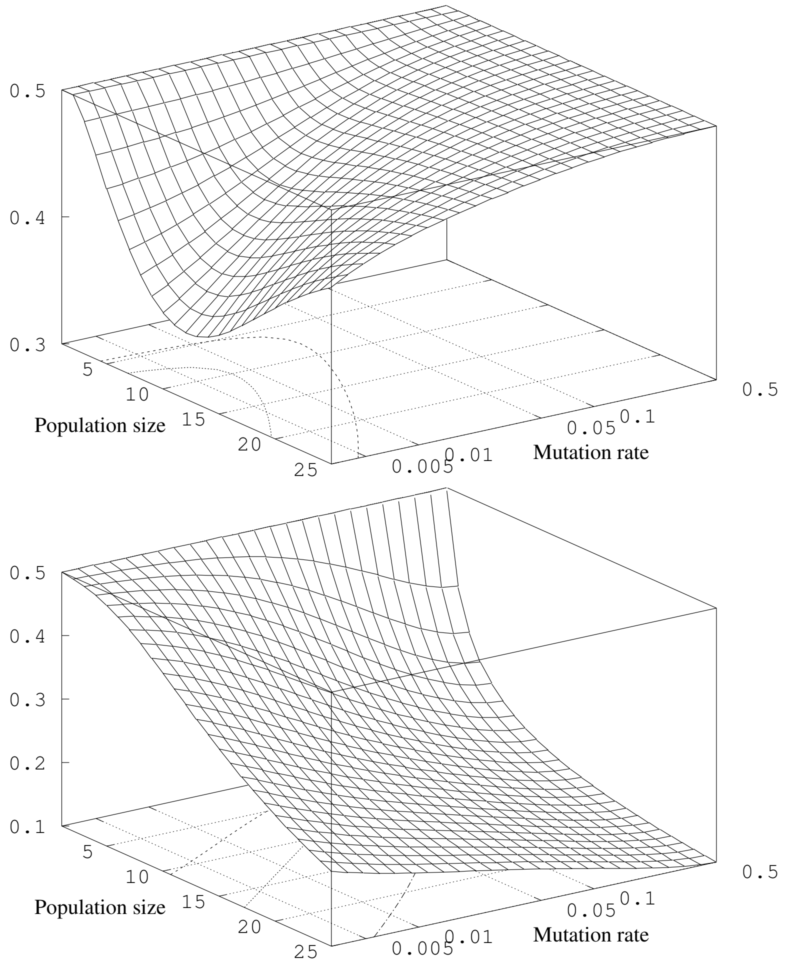


Figure 6.3: Expected proportion of Hawks (top) and standard deviation (bottom) in the limit distribution of the Hawk-Dove game, for different population sizes and mutation rates.

can force the expected behavior of the finite population model relatively far away from the infinite population's predicted stable fixed point, toward higher proportions of the Dove strategy. As the fixed points, means and standard deviations of the systems have been determined by exact techniques, it is clear that genetic drift can introduce significantly different behavior when finite population sizes are considered as compared to the predicted ESS at $(1/2, 1/2)$. These predictions lose significance as population size r or mutation rate μ is increased. Figures 6.2 and 6.3 show the influences of varying population size and mutation rate on the expected proportions of Hawk individuals in the limit distribution, and the standard deviations thereof. In comparison with Figures 4.2 and 4.3 on pages 63, 64, it is clear that the selective pressure toward the ESS in the Hawk-Dove game – selective pressure which is not present in the Neutral game – leads to smaller standard deviations around the expected proportions. Indeed, the stochastic effects of genetic drift are weaker if selective pressure is stronger.

Schaffer (1988) has proposed a correction to the ESS for the Hawk-Dove game. In this correction, Schaffer concludes that the proportion of Hawks in a finite population is higher for finite populations as compared with the ESS. This is clearly in contrast with our exact predictions of the limit behavior of the Hawk-Dove game. Note that Schaffer studies populations that are not generational, i.e., only a small part of the population is replaced by offspring individuals. Fogel et al (Fogel & Fogel 1995; Fogel, Fogel, & Andrews 1997; Fogel, Andrews, & Fogel 1998) empirically study the influence of finite populations on the stability of Hawk-Dove, with varying offspring population sizes. From their simulations, it can be observed that the bias of deviation from the ESS is indeed dependent on the size of the offspring population. The conclusions made here are thus only true for generational evolutionary systems, whose offspring population is of size r. The Markov model may be adapted to include smaller offspring populations, but this is out of scope for the text.

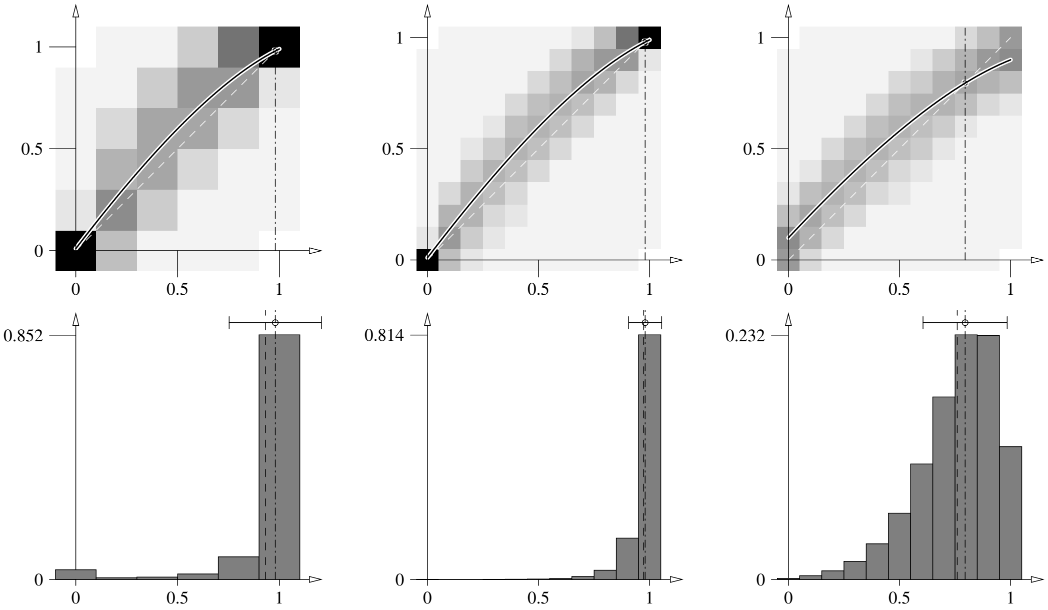
We need an explanation why small populations drift to higher proportions of Dove. Consider two finite systems, with the same population size r and mutation rate μ. The first system is initialized with a population with $r/2 - k$ Hawks, the other is initialized with $r/2 + k$ Hawks, with k strictly positive. The probability of moving from these initial states to the state with a proportion of 1/2 Hawks in n steps can be computed. The probability of reaching this state is higher when starting with $r/2 + k$ Hawks, where the system started with more Doves remains longer stuck. On average, the overall system thus remains longer in states that have a higher proportion of Doves. On average, the system is thus pushed to higher proportions of Doves as compared to the infinite model. The observation that populations drift toward higher proportions of Dove (and not the other way) is similar to Ficici and Pollack's (2000a) observation of this effect in simulation runs of the Hawk-Dove game.

As a conclusion, we thus expect that in a finite population, with a small mutation rate, we expect more cooperation (Dove strategy is more cooperative in sharing payoff) in the Hawk-Dove game, as compared with the ESS. By introducing stochastic elements, such as mutations and a small population size, the system becomes biased toward a strategy that is not evolutionarily stable, and thus contradicts the expectations of the infinite population model. With small populations, the concept of an ESS becomes obsolete.

6.5 Prisoners' Dilemma

6.5.1 Game

The well-known and broadly studied Prisoners' Dilemma originated from the following setting. Two criminals are arrested under the suspicion of a crime



(a) $r = 5, \mu = 0.01$ (b) $r = 10, \mu = 0.01$ (c) $r = 10, \mu = 0.1$

Figure 6.4: Limit or steady state distributions for the Prisoners' Dilemma game for 3 different parameter settings. The horizontal axis represents the proportion of Defect genomes in the population.

they have committed. The police doesn't have enough proof to convict them. The criminals are separately questioned. Both criminals must choose to either Cooperate with each other or to Defect. If either one of the criminals gives the police more evidence to convict the other, this Defector is freed. If both players Cooperate, they receive only a short time in jail. If both players tell out on each other, then the police has enough evidence to convict both. If one player Defects her Cooperating opponent, the Defector receives a high payoff, and the Cooperator spends a long time in jail.

Biological examples of the Prisoners' Dilemma can be found in the behavior of bacteriophage $\Phi6$ and the evolution of ATP producing pathways (Nowak & Sigmund 2004).

Consequently, the game can be modeled with two strategies $\Omega = \{C, D\}$ (Cooperate and Defect) with a payoff matrix A where $A_{C,D} < A_{D,D} < A_{C,C} < A_{D,C}$. Both pure strategies are equilibrium strategies if no variation is assumed. The fixed point where the whole population adopts Defection is stable and the ESS of the game. The fixed point where all players Cooperate is unstable. There is no mixed strategy ESS for this game.

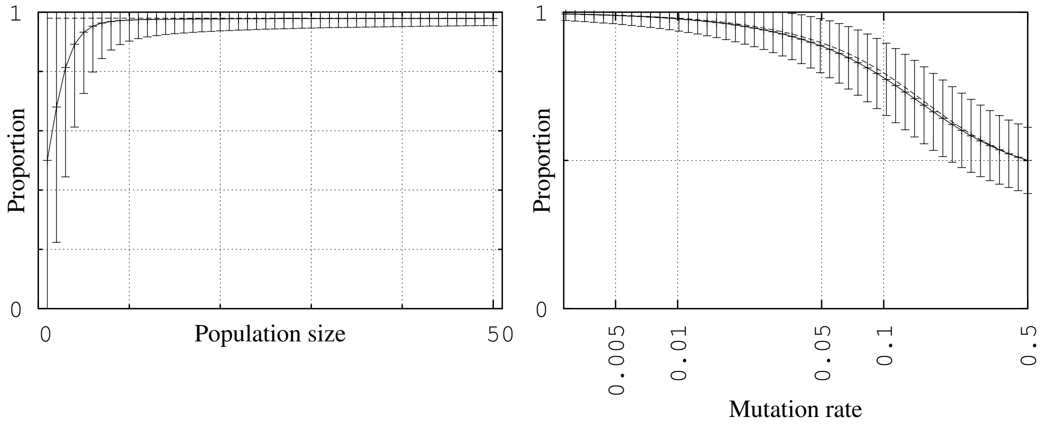


Figure 6.5: Expected proportion of Defect and standard deviation in the limit distribution of the Prisoners' dilemma, for different population sizes with $\mu = 0.01$ (left) and a range of mutation rates with $r = 20$ (right).

6.5.2 Results

We have assumed payoff values $A_{C,D} = 1 < A_{D,D} = 2 < A_{C,C} = 3 < A_{D,C} = 4$, with payoff matrix

$$A = \begin{pmatrix} 2 & 4 \\ 1 & 3 \end{pmatrix}. \tag{6.12}$$

We expect the populations to contain a lot of the Defect genomes on the long run, even if variation is assumed. The effects of genetic drift observed in the previous games are different in the Prisoners' Dilemma. In the previous games, selective pressure pulls the populations toward diverse populations, as the stable ESS of those games is a mixed strategy. At the other end, genetic drift pushes instantiations of the system to less diverse populations, such that one strategy becomes prominently abundant in the population. In the Prisoners' Dilemma, the behavior in the finite model is expected to concentrate on populations with one strategy (Defect). Since genetic drift drives populations to become fixated, we expect that in this game, selection and drift work together and consequently, that the finite model better resembles the infinite model as compared with the Hawk-Dove game.

Figure 6.4 depicts steady state distributions of the finite population model when the individuals are involved in the Prisoners' Dilemma, for a number of parameter settings. As predicted by the infinite model, the distributions are expected to have a large proportion of Defect genomes. Figures 6.5 and 6.6 show the influence of population size and mutation rate on the expectations of the finite model and the standard deviation thereof.

As the population size is increased, the expected behavior of the system better resembles the expected infinite population behavior, and the standard deviation σ around the weighted mean of the expected distribution over the states decreases.

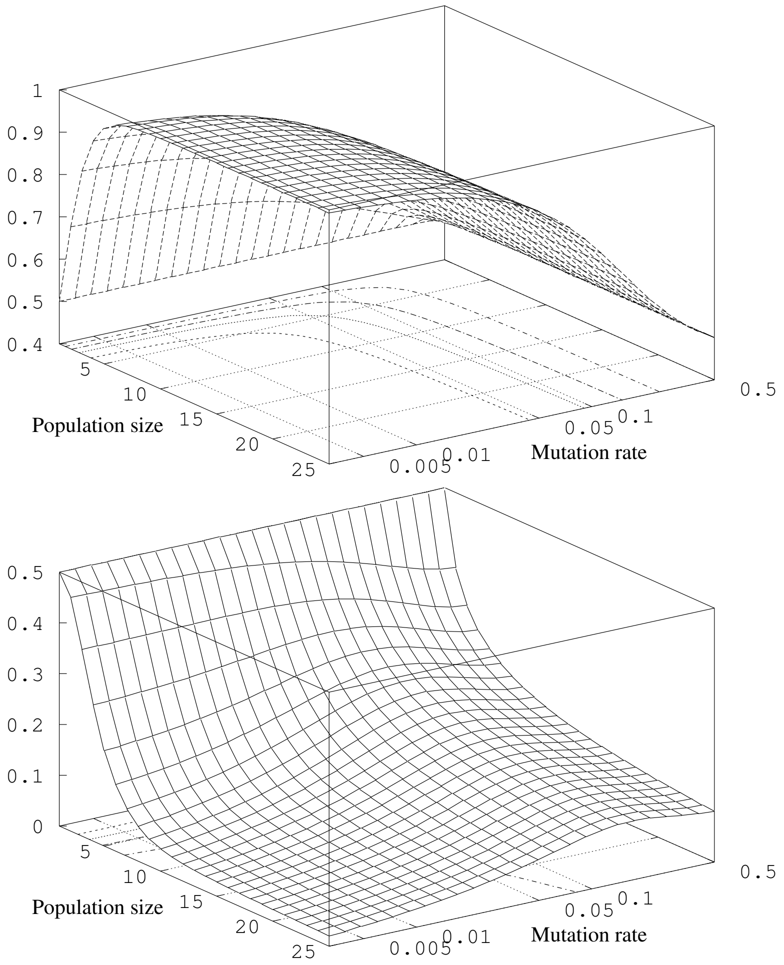


Figure 6.6: Expected proportion of Defect (top) and standard deviation (bottom) in the limit distribution of the Prisoners' dilemma, for different population sizes and mutation rates.

Only for extremely small population sizes, the effects of random sampling are clearly visible and may lead to noticeable proportions of Cooperating populations.

As we increase the rate of mutation, σ increases, as the generation of random individuals tends to push the populations to more diverse configurations. In the case of the Prisoners' Dilemma, selection pushes the population to less diverse populations. Note that this observation contrasts with the expected behavior in the previous games, where higher variation resulted in predictions that better resembled the infinite population model. Of course, in these other games, selective pressure and a high mutation rate both guide the system to more diverse populations. The expectations of the Prisoners' Dilemma game are similar to the behavior observed with a static fitness function in section 4.3. For small population sizes and small mutation rates, the predictions of the finite population model are thus more similar to the infinite model for the Prisoners' Dilemma as compared to the influence of genetic drift on the Neutral and Hawk-Dove game.

In the Prisoners' Dilemma, the Cooperate strategy is rationally the optimal strategy, if all other players in the game also opt for this strategy. In an evolutionary system, however, this pure strategy is an unstable fixed point. Only for small population sizes and extremely small mutation rates, the finite population model predicts a noticeable proportion of Cooperate genomes in the populations. Figure 6.4(a) gives an example of a small probability of ending up in a population filled with the Cooperate genome.

6.6 Coordination Game with Risk Dominance

6.6.1 Game

The last 2 × 2 game studied in this chapter is rather artificial by nature, and is inspired by the notion of risk dominance as studied extensively by Harsanyi & Selten (1988). We assume that the strategies are given by $\Omega = \{0, 1\}$. We assume that the following properties hold for the payoffs of this game:

- $A_{0,0} > A_{1,0}$ and $A_{1,1} > A_{0,1}$, i.e., the game is a coordination game. Players in the game have to agree on one of the two strategies in order to receive a high payoff. If the players do not agree, a lower payoff is rewarded. An example of a coordination game is the setting where two technologies are available to two firms with compatible products, and they have to elect a strategy to become the market standard. If both firms agree on the chosen technology, high sales are expected for both firms. If the firms do not agree on the standard technology, few sales result. Both strategies are evolutionarily stable strategies of the game.

- $A_{0,0} > A_{1,1}$, i.e., the payoff received when both players are playing strategy 0 is higher than the setting where both players adopt strategy 1. Strategy 0 is said to payoff-dominate strategy 1.

- $A_{0,0} + A_{0,1} < A_{1,0} + A_{1,1}$, i.e., an individual adopting strategy 1 has a higher expected payoff than an individual adopting strategy 0 when faced with an individual playing either strategy with probability 0.5. Since $A_{0,0} > A_{1,0}$ and $A_{1,1} > A_{0,1}$, strategy 1 is said to risk-dominate strategy 0. The loss in payoff if the opponent changes its strategy is lower for an individual playing the strategy that risk dominates the other strategy. The risk dominating strategy thus has the lowest risk of losing payoff. A risk dominating strategy is advised if an opponent is known to change its strategy.

Note that, besides the notion of dominance in diploid individuals, two extra concepts that involve the word "dominance" have been introduced. To make the difference among these concepts clear, we always use "payoff dominance" and "risk dominance" to denote the newly introduced terms, while "dominance" is used to denote the dominance of one allele over another in diploid individuals.

Players thus must coordinate their strategies, and choose either the risk dominating strategy 1 or the payoff dominating strategy 0. Both strategies are evolutionarily stable strategies of the game, and have their separate advantages. If the players are unstable in their choices, the risk dominant strategy would be preferred, and if players can trust each other, they can cooperate with the payoff dominant strategy.

Observations made by Robson & Vega-Redondo (1996) have shown that, for the limit where the mutation rate goes to 0, stochastic stability of the payoff dominating strategy is found. Miekisz (2005a) has studied some models of finite populations of the Coordination game with a risk dominant and payoff dominant strategy. By making assumptions to simplify the Markov models, his analysis shows that for an arbitrarily low mutation rate, populations that are sufficiently large have a high probability of choosing the risk dominating strategy. Miekisz concludes that, in order to describe the limit behavior of a population, the mutation rate and population size have an important impact on the expected behavior. This concurs with the conclusions of previous models and games. We carry out such a study by performing an experimental analysis of the limit behavior for the Coordination game, for specific payoffs, mutation rates and population sizes.

6.6.2 Results

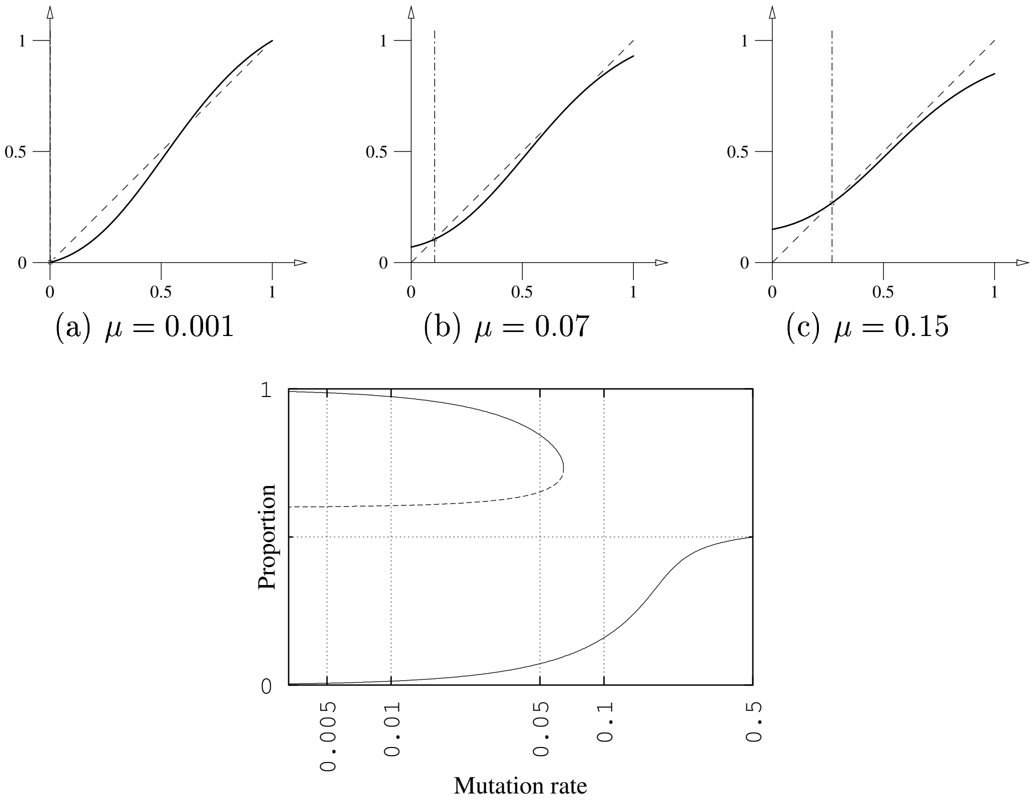
Consider the Coordination game with payoff dominance and risk dominance with payoffs according to matrix A, with

$$A = \begin{pmatrix} 5 & 1 \\ 3 & 4 \end{pmatrix}. \tag{6.13}$$

Note that these payoffs are according to the properties as itemized above. The game is a coordination game as $A_{0,0} = 5 > A_{1,0} = 3$ and $A_{1,1} = 4 > A_{0,1} = 1$. Strategy 0 payoff dominates strategy 1 as $A_{0,0} = 5 > A_{1,1} = 4$. Strategy 1 risk dominates strategy 0, since the expected payoff – against an opponent playing each strategy with probability 0.5 – is 3 for strategy 0 and 3.5 for strategy 1.

Infinite population model. In contrast with the previous games studied in this chapter, this games constitutes of 2 evolutionarily stable strategies. Dependent on the mutation rate, the infinite population model has one or two stable fixed points related to the evolutionary stable strategies. Figure 6.7 depicts heuristic $\mathcal{G}$ for specific values of μ and the stable and unstable fixed points of an infinite model involved in this game.

As the mutation rate increases, the fixed point related to the payoff dominant strategy dissolves, as is the unstable fixed point. In the limit where $\mu \to 0$, both



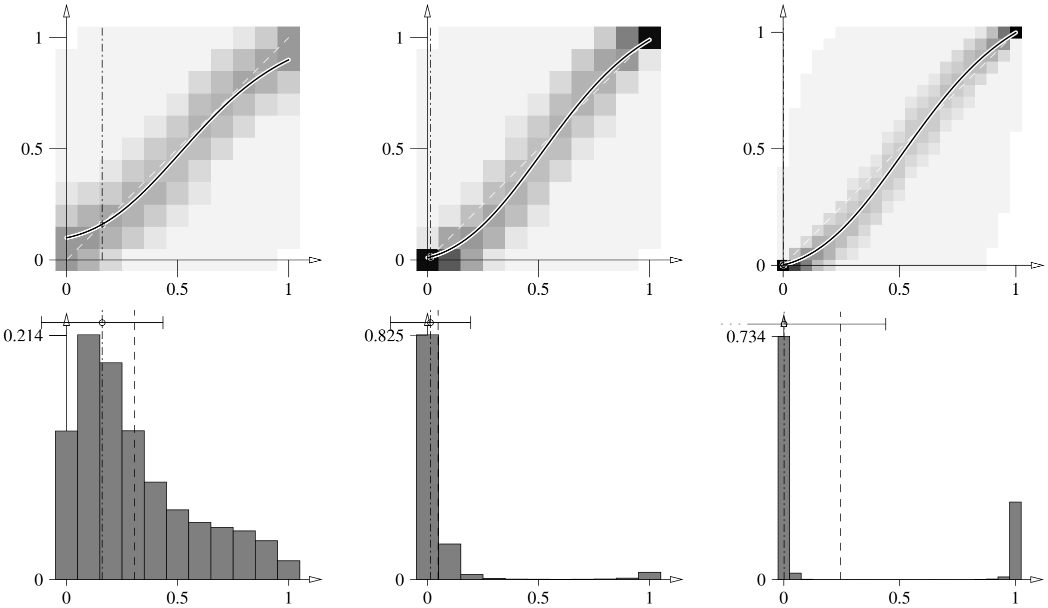
(a) $\mu = 0.001$ (b) $\mu = 0.07$ (c) $\mu = 0.15$

Figure 6.7: Top: Heuristic $\mathcal{G}$ of the Coordination game with payoff and risk dominance for varying mutation rates μ. Axes denote proportions of payoff dominance in the current and next population. The fixed point to which a population started in $(1/2, 1/2)$ is attracted is shown. Bottom: Fixed points of the game for different mutation rates. The vertical axis represents proportions of payoff dominant strategies in the population. The solid graphs represent stable fixed points, where the dashed graph represents unstable fixed points.

states whose proportion of payoff dominant strategies equals 0 and 1 are stable fixed points, and 0.6 is the unstable fixed point of the system. This unstable fixed point is given, in similarity with the mixed strategy ESS of the Hawk-Dove game, by (6.9).

Consider a sufficiently small mutation rate, such that the unstable fixed point exists. If the initial proportion of payoff dominant strategies in the initial infinite population is high enough (i.e., higher than the proportion of payoff dominant strategies of the unstable fixed point), the deterministic system is attracted by the stable fixed point related to payoff dominance. In all other initial cases, the system moves to the stable fixed point related to risk dominance.

These observations of expected behavior are according to our previous expectations. The individuals become less stable in their choice of strategy as the mutation rate increases, and consequently, the population becomes more likely to opt for the risk dominating strategy. This is also of influence on the behavior in finite population models.



(a) $r = 5, \mu = 0.1$ (b) $r = 10, \mu = 0.01$ (c) $r = 20, \mu = 0.001$

Figure 6.8: Transition matrices and limit or steady state distributions for the Coordination Game with Risk Dominance for 3 different parameter settings. The horizontal axis represents the proportion of payoff dominant strategies in the population.

Note that in the comparisons of the limit behavior of the finite population model with that of the infinite model, we only consider the fixed point related to the risk dominating strategy. This is due to the fact that we assume that the initial infinite population that starts the deterministic chain is given by the population whose ratio among the two frequencies equals $1 : 1$. The system is attracted by the fixed point related to the risk dominating strategy if started in this initial population, independent of the mutation rate, since the proportion of payoff dominance of the unstable attractor in the infinite model is always higher than $1 : 1$.

Finite population model. Figure 6.8 depicts the transition probability matrices and limit distributions for some particular instantiations of the finite population model. Figures 6.9 and 6.10 relate population size and mutation rate to expected proportions, and standard deviations thereof, of the payoff dominating strategy.

The probability to end up with a population of risk dominating strategies is generally high, as predicted by the infinite population model. As populations become very small, the influence of genetic drift overwhelms the selective pressure in the game, which causes the populations to be in either the state where they consist of strategy 0 and 1 only, as before in other games.

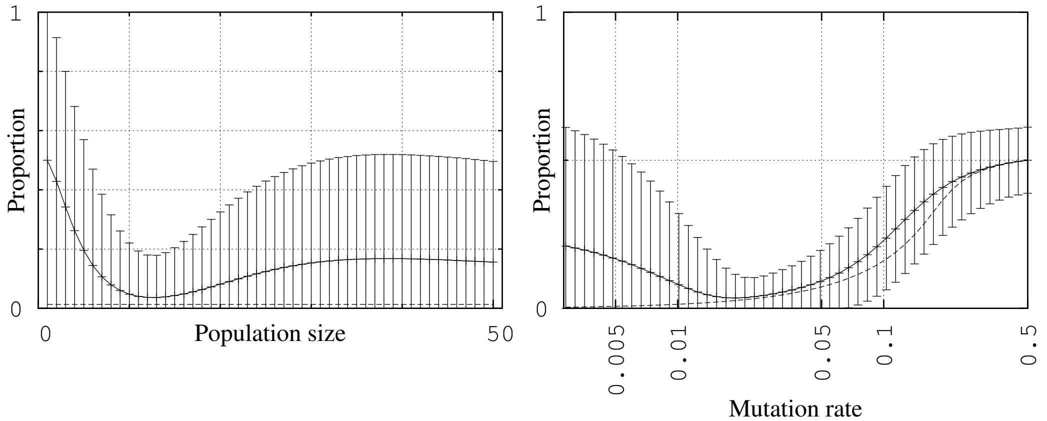


Figure 6.9: Expected proportion of payoff dominant strategies and standard deviation in the limit distribution of the Coordination game, for different population sizes with $\mu = 0.01$ (left) and a range of mutation rates with $r = 20$ (right).

For large mutation rates, i.e., close to 0.5, the mutation rate is high enough to, again, overpower the selective pressures present in the game, in similarity with the predictions of high mutation rates in the infinite population model and mutation rates.

As expected, in relatively large populations with a small mutation rate, the individuals are more confident of the stability of their opponents to assume a slightly larger probability of adopting the payoff dominating strategy. See for an example the limit distribution with $r = 20, \mu = 0.001$ in Figure 6.8(c). As the population size approaches the limit of infinitely large populations, however, the limit behavior returns to the predicted, lower, proportions of payoff dominance in the infinite model.

In between these observations, we are left with systems with a moderate population size in relation to the mutation rate. In this case, the system is not under large influence of genetic drift, nor of being attracted to highly diverse populations. On the other hand, the populations are not large enough in relation to the mutation rates that individuals can take the risk to opt for the payoff dominating strategy, as the probability that individuals in the population are mutated to the risk dominating strategy is too high. For these parameter settings, the expected proportion of risk dominating strategies is thus highest. An example of the steady state distribution of the system for such parameters is given with $r = 10, \mu = 0.01$ in Figure 6.8(b).

For the Coordination game, we can consequently conclude that the size of the population and the rate of mutations in the evolutionary system have an important impact on the expected behavior of the population. The individuals in the population adopt the risk dominating strategy if the stochastic elements of the evolutionary process are prevalent. Only in the case where the population may remain sufficiently stable and is free of stochastic distortions, the payoff dominating strategy has a relatively higher probability of being adopted. The

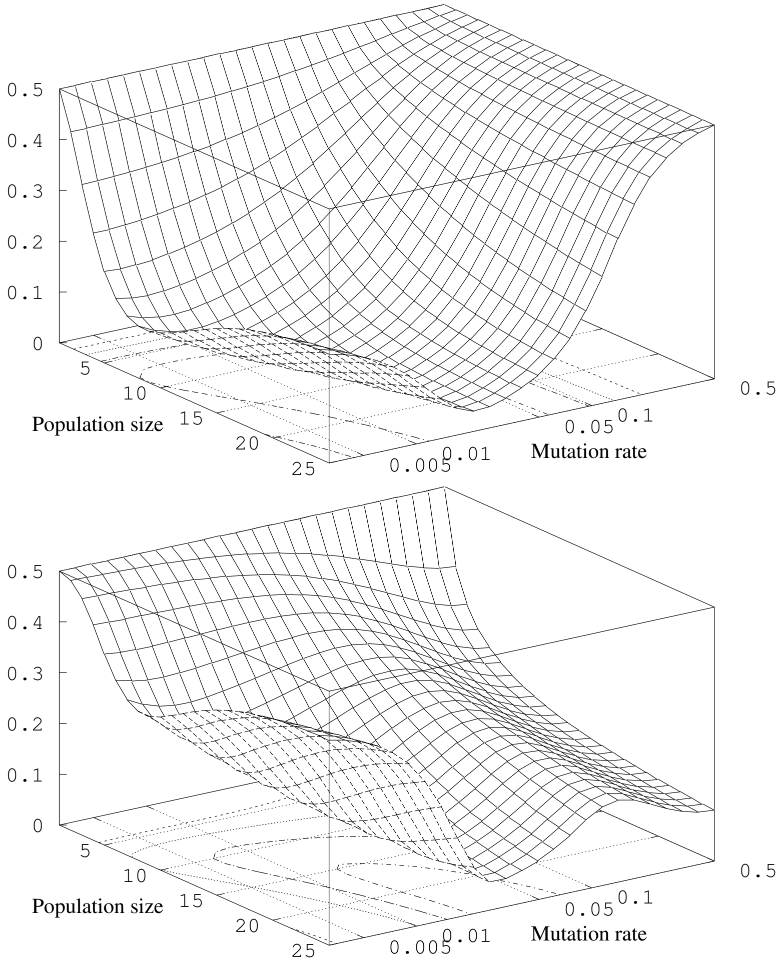


Figure 6.10: Expected proportion of payoff dominant strategies (top) and standard deviation (bottom) in the limit distribution of the Coordination game, for different population sizes and mutation rates.

amount of stochastic events – caused by the finiteness of the population and the amount of variation – thus has an important influence on the choice of strategy that is adopted by the finite population.

6.7 Rock-Paper-Scissors

6.7.1 Game

In the well-known Rock-Paper-Scissors game, two players have the choice of three strategies, i.e., $\Omega = \{R, P, S\}$ (denoting strategies Rock, Paper, Scissors, respectively). In this game, Paper beats Rock, Rock beats Scissors, and Scissors in its turn beats Paper. If both players choose the same strategy, there is a tie in

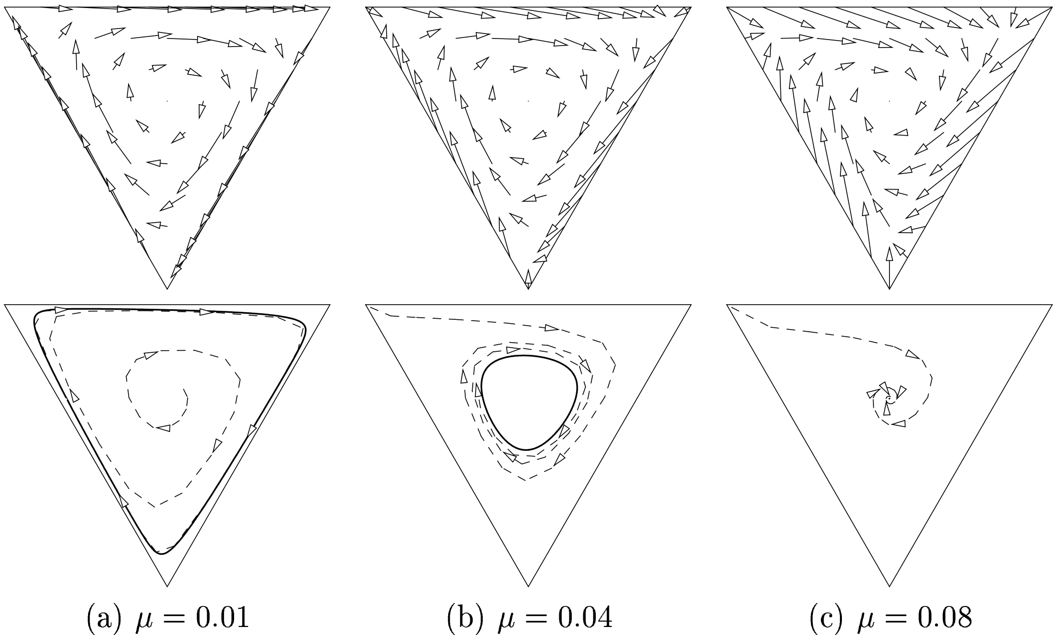


Figure 6.11: Transient behavior of the infinite population model for Rock-Paper-Scissors. Each triangle represents the state space of populations for the Rock-Paper-Scissors game; the top left corner of the triangle represents a population with all Scissors, the top right corner Rock, and the bottom corner Paper. Top: Vector fields depicting the mapping of the state space. Bottom: Example traces of transient behavior and limit cycle of the infinite population model in bold, if available. In (a), the chain starts with proportions 0.4 Rock and 0.3 Paper, and approaches a limit cycle from within. In (b), the initial proportions are 0.01 Rock and 0.01 Paper, and approaches a limit cycle from the exterior. In (c), the initial proportions of strategies are as in (b), this chain converges to a stable fixed point at the center of the state space.

payoff. We can write the game's payoff matrix A with

$$A = \begin{pmatrix} t & l & w \\ w & t & l \\ l & w & t \end{pmatrix} \tag{6.14}$$

with payoffs according to $l < t < w$, denoting the payoffs received in case of a loss, tie or win, respectively.

6.7.2 Dynamics

For our instantiation of the Rock-Paper-Scissors game, assume that the payoffs received are according to $l = 0 < t = 1 < w = 2$. Winning a game results in a payoff of 2, playing a tie results in payoff 1 for each player, and a loser gains no points. In a population that mainly consists of individuals adopting strategy Paper, the individuals with strategy Scissors receive the highest payoff, and their proportion in the population increases. Once their proportion has become

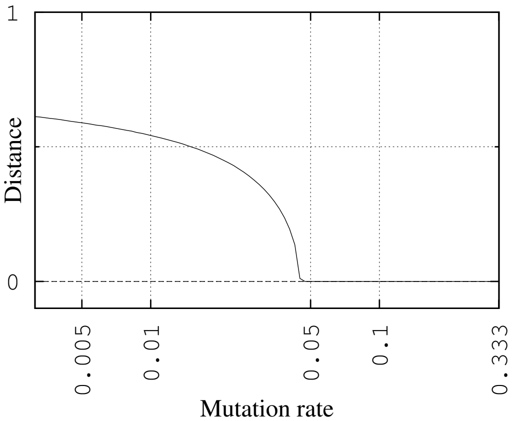


Figure 6.12: Mean distance between the states that constitute the limit cycle, on one hand, and the fixed point at $(1/3, 1/3, 1/3)^\top$, on the other hand, for different mutation rates.

relatively large, the expected payoff of the Rock strategy becomes highest, and the proportions of individuals adopting this strategy starts to increase. Once the population merely consists of Rock, it is again taken over by the Paper strategy.

A population that constitutes only of the Rock strategy, can be invaded by a mutant Paper strategy, alike for Scissors invading Paper and Rock invading Scissors. Each of the pure strategies is thus – in the case of infinite population size and no variation – an unstable fixed point of the model. Since none of the strategies can persist without being invaded by a mutant strategy, there is no ESS of the game. A fourth, mixed strategy fixed point in the infinite model exists at $(1/3, 1/3, 1/3)^\top$ as the game is symmetric for either strategy. The discrete time behavior of a population engaged in the Rock-Paper-Scissors game cycles around the center of the state space with proportions $(1/3, 1/3, 1/3)^\top$ of the population state space. Mappings of the state space and example trajectories of an infinite population in the Rock-Paper-Scissors game are shown in Figure 6.11.

For small mutation rates, the infinite model is attracted to states with high proportions of either strategy. For higher mutation rates, the population is maintained highly diverse, and is attracted to states that are closer to the fixed point at $(1/3, 1/3, 1/3)^\top$. For small values of μ, the infinite model in discrete time approaches a limit cycle*, i.e., the fixed point at $(1/3, 1/3, 1/3)^\top$ is unstable. For larger mutation rates, the fixed point at $(1/3, 1/3, 1/3)^\top$ is observed to be stable. Figure 6.12 shows the mean distance of the limit cycle of the infinite model to the fixed point at $(1/3, 1/3, 1/3)^\top$ for varying mutation rates. For very small mutation rates, the limit cycle is close to the border of the state space. As the mutation rate is increased, the limit cycle is closer to the center of the state space. As the mutation rate approaches approximately 0.045, any instantiation of the

*In the case of the discrete time model, we cannot really speak of a limit cycle as the model takes discrete steps in the state space. The infinite model approaches a set of states and, in the limit, remains within this cyclic set of states. We thus use the term *limit cycle* as a figure of speech to denote this set, which resembles a continuous limit cycle infinite model spirals toward the fixed point at the center, thus becoming the stable fixed point of the system. We can determine the exact value of μ where the stability property of the fixed point at $(1/3, 1/3, 1/3)^\top$ changes.

Let (p_R, p_P, p_S) denote a population with its respective proportions of Rock, Paper and Scissors. Since $p_S = 1 - p_R - p_P$, we can write a population as a vector $\mathbf{p} = (p_R, p_P)$ and the dynamics of the infinite model in proportions of Rock and Paper only.

The fitness of individuals adopting Rock, Paper and Scissors is given by

$$\begin{array}{rclcl} f(R, \mathbf{p}) & = & p_R + 2p_S & = & 2 - 2p_P - p_R \\ f(P, \mathbf{p}) & = & p_P + 2p_R & & \\ f(S, \mathbf{p}) & = & p_S + 2p_P & = & 1 + p_P - p_R \end{array}$$

and the mean fitness of all individuals in the whole population always equals

$$\overline{f}(\mathbf{p}) \;=\; p_R f(R, \mathbf{p}) + p_P f(P, \mathbf{p}) + (1 - R - P) f(S, \mathbf{p}) \;=\; 1.$$

The probabilities that either strategy is selected from population $\mathbf{p}$ is given by

$$\begin{array}{rclcl} S(R, \mathbf{p}) & = & p_R(2 - 2p_P - p_R) & & \\ S(P, \mathbf{p}) & = & p_P(p_P + 2p_R) & & \\ S(S, \mathbf{p}) & = & (1 - p_R - p_P)(1 + p_P - p_R) & = & 1 + p_R^2 - 2p_R - p_P^2. \end{array} \qquad (6.15)$$

According to a mutation rate μ, we can write the elements of the heuristic with

$$\begin{array}{rcl} G(R, \mathbf{p}) & = & (1 - 2\mu) S(R, \mathbf{p}) + \mu(S(P, \mathbf{p}) + S(S, \mathbf{p})) \\ G(P, \mathbf{p}) & = & (1 - 2\mu) S(P, \mathbf{p}) + \mu(S(R, \mathbf{p}) + S(S, \mathbf{p})) \\ G(S, \mathbf{p}) & = & 1 - G(R, \mathbf{p}) - G(P, \mathbf{p}). \end{array} \qquad (6.16)$$

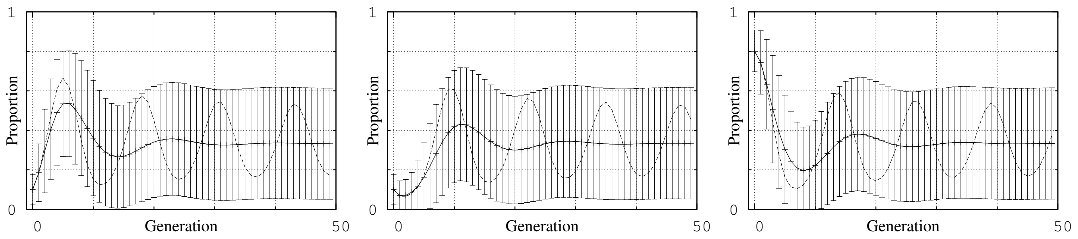
which can be expanded to give the dynamics of heuristic $\mathcal{G}$, expressed in proportions of Rock and Paper only, by

$$\begin{array}{rclcl} \mathcal{G}_R(\mathbf{p}) & = & G(R, \mathbf{p}) & = & p_R(2 - 2p_P - p_R) + \mu(1 - 6p_R + 6p_P p_R + 3p_R^2) \\ \mathcal{G}_P(\mathbf{p}) & = & G(P, \mathbf{p}) & = & \mu + (1 - 3\mu)(p_P^2 + 2p_P p_R) \end{array} \qquad (6.17)$$

for which $\hat{\mathbf{p}} = \mathcal{G}(\hat{\mathbf{p}}) = (1/3, 1/3)^\top$ is a fixed point. We now want to determine for which values of μ this fixed point is stable. In order to do so, we need to determine for which values of μ the Jacobian of (6.17), evaluated at the fixed point, has eigenvalues λ such that $|\lambda| < 1$.

The Jacobian of $\mathcal{G}$ is equal to

$$\begin{pmatrix} \frac{\partial \mathcal{G}_R}{\partial p_R} & \frac{\partial \mathcal{G}_R}{\partial p_P} \\ \frac{\partial \mathcal{G}_P}{\partial p_R} & \frac{\partial \mathcal{G}_P}{\partial p_P} \end{pmatrix} = \begin{pmatrix} 2 - 2p_P - 2p_R + 6\mu(-1 + p_P + p_R) & -2p_R + 6\mu p_R \\ 2(1 - 3\mu)p_P & 2(1 - 3\mu)(p_P + p_R) \end{pmatrix},$$



Rock Paper Scissors

Figure 6.13: Expected proportions and standard deviations of Rock, Paper and Scissors strategies of first 50 generations, for a finite population with $r = 25, \mu = 0.04$, and respective proportions in the infinite population (dashed) with $\mu = 0.04$. Both models have been seeded with initial frequencies of Rock, Paper and Scissors strategies 0.1, 0.1 and 0.8, respectively.

which, evaluated at $\hat{\mathbf{p}} = (1/3, 1/3)^\top$ gives

$$\begin{pmatrix} \frac{2}{3} - 2\mu & -\frac{2}{3} + 2\mu \\ \frac{2}{3} - 2\mu & \frac{4}{3} - 4\mu \end{pmatrix}.$$

The eigenvalues λ of this matrix are given by the solution of characteristic equation

$$\lambda^2 + (6\mu - 2)\lambda + 12\mu^2 - 8\mu + 4/3 = 0$$

with solutions

$$\lambda = -3\mu + 1 \pm \sqrt{3}i \left| \mu - \frac{1}{3} \right|.$$

The infinite population model's fixed point at $(1/3, 1/3)^\top$ is stable if $|\lambda| < 1$. We compute the value of μ for which $|\lambda|^2 = 1$, i.e., where

$$\begin{aligned} 1 &= |\lambda|^2 \\ &= (-3\mu + 1)^2 + 3(\mu - 1/3)^2 \\ &= 12(\mu - 1/3)^2 \end{aligned}$$

holds, which is the case if mutation rate μ equals

$$\mu = \frac{1}{3} \pm \frac{1}{6}\sqrt{3}.$$

The infinite population model's fixed point $\hat{\mathbf{p}}$ at $(1/3, 1/3, 1/3)^\top$ is stable if the mutation rate μ is according to $\frac{1}{3} - \frac{1}{6}\sqrt{3} < \mu < \frac{1}{3} + \frac{1}{6}\sqrt{3}$, which is approximately $0.0446582 < \mu < 0.622008$. Note that we are mainly interested in mutation rates that are according to $\mu < 1/3$.

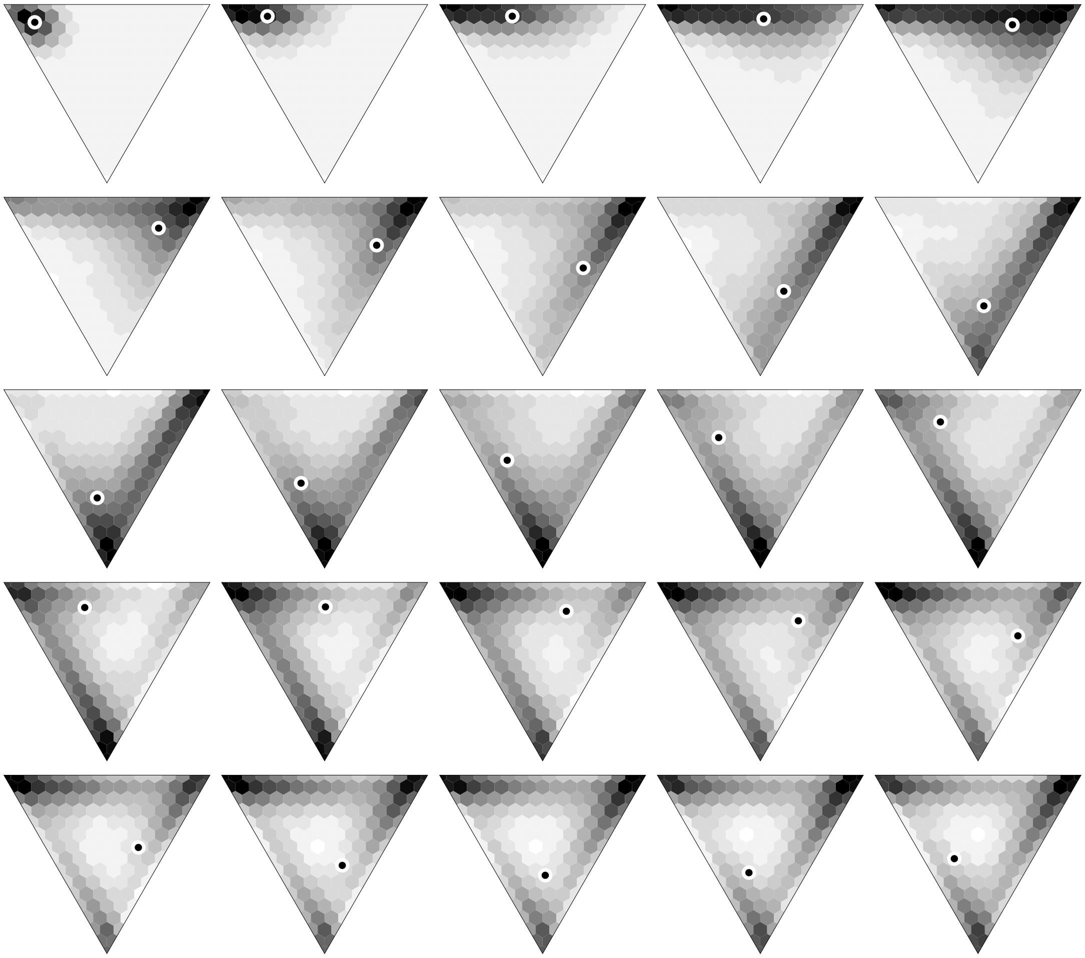


Figure 6.14: Distributions of first 25 generations, showing the transient behavior of Rock-Paper-Scissors for a finite population with $r = 25, \mu = 0.04$, with initial frequencies of Rock, Paper and Scissors strategies 0.1, 0.1 and 0.8, respectively. A darker gray scale represents a higher probability of being in that state. The superimposed dot represents the transient behavior of the corresponding infinite population model.

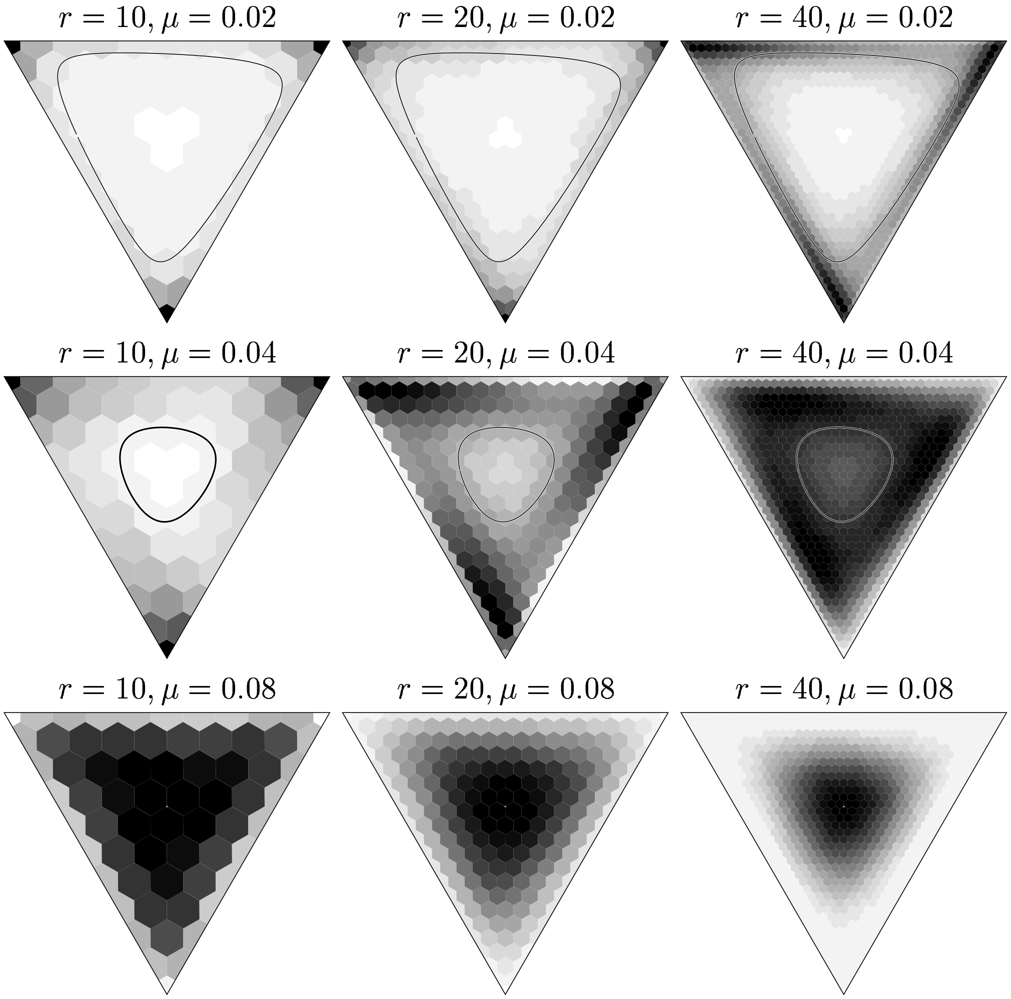
6.7.3 Finite model

Transient behavior. Figures 6.13 and 6.14 show the transient behavior of the infinite model, and distributions of the finite model for the initial generations, if the model is initialized with initial probabilities $(0.1, 0.1, 0.8)^\top$ for the respective strategies. In the finite population model, a population size of $r = 25$ was chosen, and the initial distribution is seeded according to (3.28). Both models assume a mutation rate of $\mu = 0.04$, i.e., the infinite system converges toward a limit cycle in the long run. We should point out that the gray scale in Figure 6.14 used for displaying the finite model is different for each of the steps, to better depict the distribution at each time step. At generation 25 the distribution of the finite model already resembles the limit distribution of the model. Thus, even for a small mutation rate, and rather large population size, the Markov model mixes relatively rapidly. We discuss the limit behavior of the finite model later.

For the transient behavior, we note that the finite model lags behind in comparison with the progress made in the infinite model. That is, if the finite population contains a high proportion of one strategy, it tends to become fixated because of the effects of random sampling. After a number of generations of fixation, the finite system rapidly moves on to states fixated at the strategy that defeated the previous prominent strategy. Because of these periods of fixation, the finite model requires more time to step from one corner of the search space to the next. The probability of fixation is high in comparison with the rate of mutants entering the population, which allows the population to escape from one fixated population and move on to the next. The deterministic model of the infinite population is not affected by these periods of stochastic fixation. Consequently, the infinite model is better able to cycle around the state space. Note that for a small search space of 3 strategies, even a population of 25 individuals would commonly be considered to be large enough to cope with the stochastic effects of fixation, and would be expected to resemble the infinite model closely. This is, however, clearly not the case.

Rapid bursts of innovation, i.e., the rapid evolution from one strategy to an other, which are intermitted by periods of fixation, as can be observed in the finite model, are the typical properties of punctuated equilibria, epochal evolution or metastability. Note that these three names in fact denote the same concept, where different authors prefer different names. Vose & Liepins (1991) and Van Nimwegen et al (1997, 1998, 1999) have observed these punctuated equilibria in the transient behavior of genetic algorithms, e.g., in the dynamics of an algorithm optimizing Royal Road functions (van Nimwegen, Crutchfield, & Mitchell 1999). They have stated that finite population models induce such punctuated equilibria in the transient behavior of the genetic algorithm. Because of the cyclic behavior of the Rock-Paper-Scissors game, where the dynamics of the game and selective pressure are repeated indefinitely, we can thus observe how this constant generation of metastability weighs in in the long run.

Distance measure. The limit behavior of the infinite model may be cyclic in a set of states in the long run. More specifically, the limit distribution of the infinite model is in this case a specific distribution over these states. Since both the infinite and finite population model now result in a limit distribution, we cannot simply compute the distance, or deviation of the limit behavior of a finite population model with respect to the infinite population model. The difficulty arises as we have to measure the distance between two distributions, in contrast with measuring the distance between a distribution (of the finite model) and a point (representing the infinite model) as in Section 3.7.2. Finding a reallocation of the probabilities from one distribution to the other, such that the



$r = 10, \mu = 0.02$ $r = 20, \mu = 0.02$ $r = 40, \mu = 0.02$

$r = 10, \mu = 0.04$ $r = 20, \mu = 0.04$ $r = 40, \mu = 0.04$

$r = 10, \mu = 0.08$ $r = 20, \mu = 0.08$ $r = 40, \mu = 0.08$

Figure 6.15: Limit distributions of finite and infinite populations involved in the Rock-Paper-Scissors game, for varying population sizes and mutation rates. In the case of $\mu = 0.02, 0.04$, the limit distribution of the infinite model converges to a limit cycle, if $\mu = 0.08$, the infinite model converges to the stable fixed point at the center of the state space.

distance over which these probabilities are reallocated is minimal would give us a suitable distance measure between the two distributions. However, we encounter some problems with such a measure. First, the distance measure by itself is an optimization problem, making it a computationally expensive measure. Secondly, the set that defines the limit cycle is not clearly defined as no proper analytic description is available. The limit cycles used in our measurements have been determined through iteration of the model, which provides a sufficiently correct approximation. In all of the experiments, the deterministic infinite model was ran for 10000 generations, after which 500 data points were collected to represent the limit cycle.

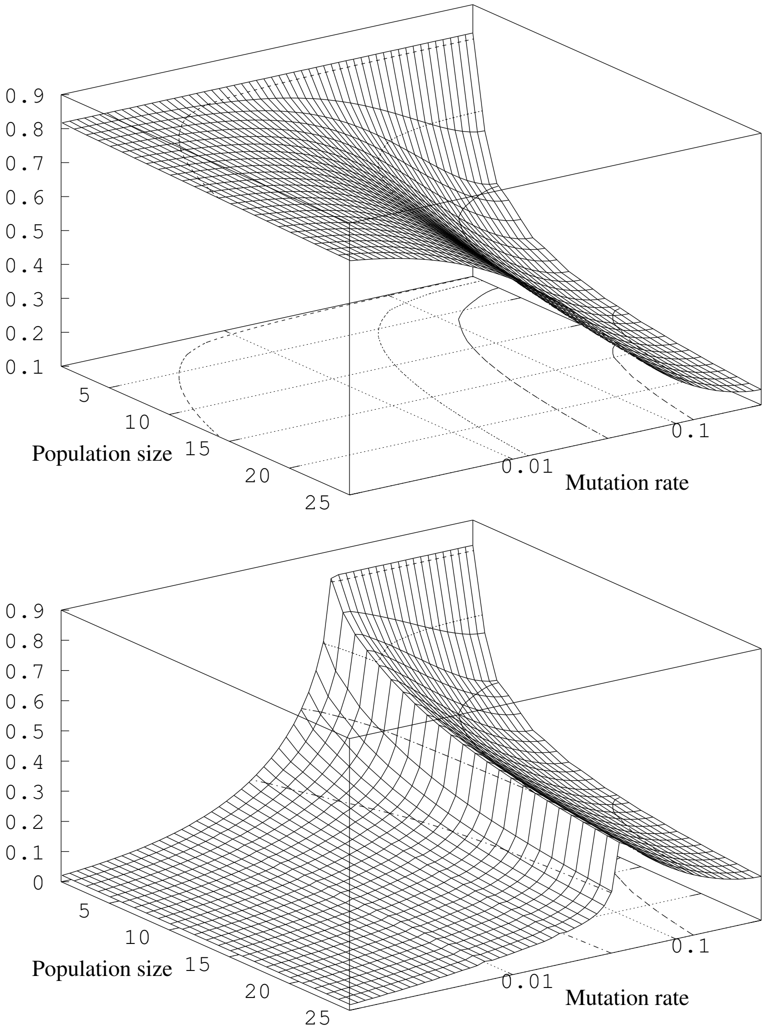
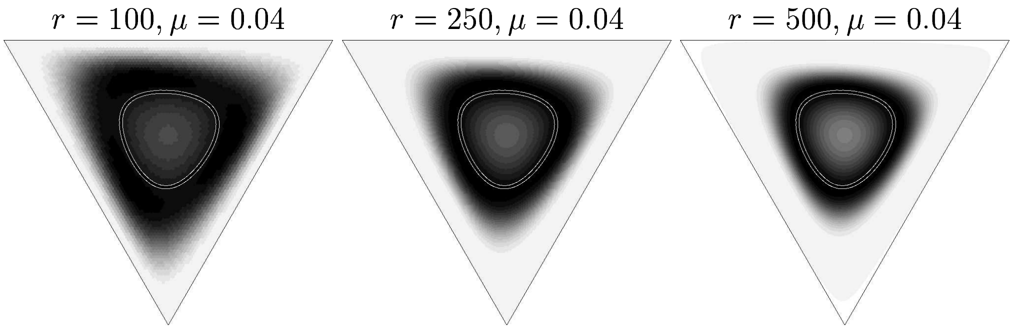


Figure 6.16: Distance of the limit distribution of the finite model to the fixed point at $(1/3, 1/3, 1/3)^\top$ (top figure) and distance to the limit cycle of the infinite model (bottom figure), for various mutation rates and small population sizes.

We can, however, provide an approximation of this distance measure that can be computed easily. For each of the populations $\mathbf{p} \in \pi$ in the finite population model, we find the closest point $\mathbf{p}_\infty$ in the infinite model's limit cycle $\mathbf{x}^*_\infty$, and weigh this Euclidean distance by the proportion of population $\mathbf{p}$ in the finite model's limit distribution $\mathbf{x}^*$. More formally, the approximation of the distance of the finite population model's limit distribution $\mathbf{x}^*$ with respect to the infinite population model's limit cycle $\mathbf{x}_\infty$ is given by

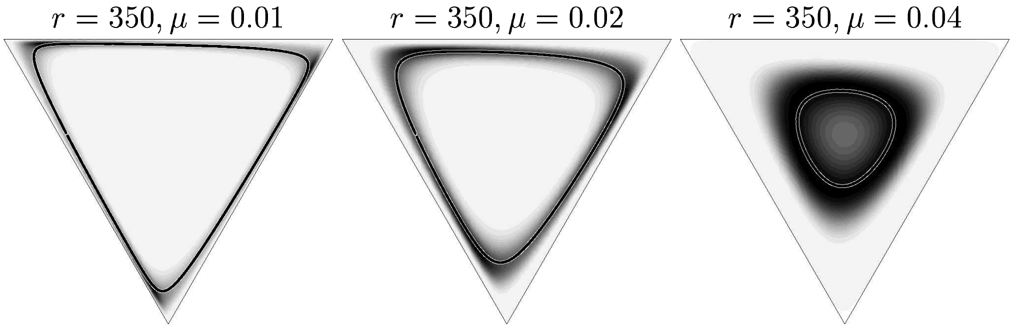
$$\sum_{\mathbf{p}\in\pi} x_{\mathbf{p}} \min_{\mathbf{p}_\infty \in \mathbf{x}_\infty} \left\{ \sqrt{(\mathbf{p} - \mathbf{p}_\infty)^2} \right\}. \tag{6.18}$$

Note that, in the case of large μ, where the limit behavior of the infinite model



$r = 100, \mu = 0.04$ $r = 250, \mu = 0.04$ $r = 500, \mu = 0.04$

Figure 6.17: Limit distributions of finite and infinite populations involved in the Rock-Paper-Scissors game, for mutation rate $\mu = 0.04$ and very large population sizes.



$r = 350, \mu = 0.01$ $r = 350, \mu = 0.02$ $r = 350, \mu = 0.04$

Figure 6.18: Limit distributions of finite and infinite populations involved in the Rock-Paper-Scissors game, for very large population size $r = 350$ and mutation rates $\mu = 0.01, 0.02, 0.04$.

does not consist of a limit cycle but of a single point at $(1/3, 1/3, 1/3)^\top$, the adopted distance is no longer an approximation, but is equal to the distance of Section 3.7.2.

Limit behavior. Figure 6.15 shows the limit distributions of the finite population model for various mutation rates and population sizes, overlaid by the limit cycle of the infinite population model. As expected from our observations of the short term behavior of the finite population, the finite models tend to reside in populations with high proportions of either strategy with a high probability, also in the long run. This effect is stronger for small population sizes and small mutation rates.

Figure 6.16 shows the distance from the limit distribution in the finite model to fixed point at $(1/3, 1/3, 1/3)^\top$ and to the limit cycle of the infinite model, as in Figure 6.18. For very small population sizes, the model has a high probability to reside in the "corners" of the search space. For small population sizes, and any mutation rate, the populations tend to become fixated in these corners, which denote populations with high proportions of one strategy. For larger mutation rates, the populations in the finite model become highly diverse, and the model resides around the infinite model's fixed point at $(1/3, 1/3, 1/3)^\top$. As mutation rates are increased and approach $1/3 - \sqrt{3}/6$, the limit behavior of the infinite population quite suddenly becomes stable at the fixed point at $(1/3, 1/3, 1/3)^\top$, where the behavior of the finite model slightly goes toward this point.

Recall Figure 6.12. As the mutation rate approaches $1/3 - \sqrt{3}/6$, the limit cycle of the infinite model rapidly goes toward to the fixed point at the center. In contrast, the finite model's predictions show that as the mutation rate is increased, its distance to the fixed point at the center goes smoothly to 0. This difference explains the ridge in the bottom figure of 6.16. At the mutation rate where the infinite model's fixed point at the center of the state space becomes stable, the difference between the finite model and the infinite model is largest. As the population size of the finite model is increased, the limit behavior of the finite model better resembles the expected behavior of the infinite population model. We expect that, as the population size of the finite model goes toward infinity, the finite model's expectations converge to the infinite population model.

Figures 6.17 and 6.18 show the limit behavior of the finite population model for large population sizes, and various mutation rates. We want to point out that the computation of the limit behavior of the model with populations of $r = 500$ individuals required our parallel implementation of the power method to determine the eigenvector of a matrix of size `118GiB`. It can be observed that for such large population sizes (relative to the model's search space of 3 strategies), the finite model's limit behavior is still visibly different from the expectations of the infinite population model. These differences are due to the constant improvement that is required in the changing fitness environment of the finite population model, and the model experiences – even for $r = 500$ – constant pressure of the stochastic sampling effects.

Summary

We have studied models of finite populations evolving strategies in the context of evolutionary game theory. The fitness functions are adapted from simple games, and render payoff or fitness dependent on the frequencies of strategies in the current population. As a result, the fitness of an individual changes over time as the composition of the evolving population changes. Because of the stochastic nature of finite population models, we question the stability of evolutionarily "stable" strategies in such models. We have studied the influence of mutation rates and finite population sizes on the expected long term behavior.

An infinite population model is attracted toward the evolutionarily stable strategies of the game, if these exist. The stability of adopting such strategies in an infinite population model is dependent on the payoffs in the game. For some games, one pure strategy may be benefecial over other strategies, such that adoption of that strategy is preferred. For other games, the population may want to adopt a mixed strategy as its optimal behavior. Games may also put forward more complex dynamics in infinite populations, such as limit cycles. By dropping the assumption of an infinitely large population, we can study how finite populations behave in the context of these games.

We can categorize two types of evolutionary dynamics in a game's fitness function. On one hand, the evolutionary dynamics of the infinite model may converge to an equilibrium situation in the long term. The population converges to an evolutionarily stable strategy, and the dynamics of the fitness function thus also converges to a static situation in the limit. In such cases, the finite model acts similarly to an evolutionary system in a static fitness environment in the long run. On the other hand, we have considered the Rock-Paper-Scissors game, where the population may continuously change because of evolutionary pressure, even toward the limit. In such perpetually dynamic situations, finite models behave similarly to models of evolutionary systems in perpetually dynamic fitness environments, as studied in the previous chapter.

In a finite population, the system balances between the forces put forth by the infinite model on one hand, and genetic drift on the other hand, which forces populations to become fixated in specific strategies. If one pure strategy is evolutionarily stable, as in the Prisoners' Dilemma game, the influence of introducing a finite population is similar to that of the models with a static selective pressure as studied in Section 4.3. If a mixed strategy is the evolutionarily stable strategy, as in the Hawk-Dove game, then genetic drift may push the population out of this mixed strategy. As a result, the population may become biased toward one strategy in comparison with the infinite model's dynamics. This allows the average behavior of the finite population to drift away from the evolutionarily stable strategy. Similarly, a specific finite population size and mutation rate may force the finite population to opt for different strategies than the ones expected in infinite population models, as in the Coordination game with Payoff and Risk dominance.

For games whose behavior is perpetually dynamic, as in the Rock-Paper-Scissors game, the effects of introducing a finite population size is most noticeable. The continuously changing dynamics of the population forces the individuals to continuously re-explore the genotype space for advantageous strategies. In similarity with the results from the previous chapter, finite populations in such dynamic environments constantly become fixated in a strategy. Because of the stochastic effect of fixation, which is not present in the infinite model, dissimilarities between the finite and infinite models emerge. A finite population evolving strategies for the Rock-Paper-Scissors game advances more slowly because of continuous occurrences of fixation. This is true even for relatively large, but finite population sizes.

Chapter 7

Co-Evolutionary Fitness Environments

In order to study evolutionary systems in co-evolutionary environments, we construct a Markov model of co-evolution of two populations with fixed, finite population sizes. In this combined Markov model, the behavior toward the limit can be utilized to study the relative performance of the evolutionary systems. As an application of the model, we perform an analysis of the relative performance of haploid and diploid genetic populations in the co-evolutionary setup, under several parameter settings.

We construct models of co-evolution of two finite populations. These models are adopted to study the limit behavior of populations involved in a competitive game, the Matching Pennies game. We provide general results of two competing haploid populations, and a setting in which a haploid population competes with a diploid population.

Parts of this chapter are derived from A.M.L. Liekens, H.M.M. ten Eikelder, P.A.J. Hilbers, *Finite Population Models of Co-Evolution and their Application to Haploidy versus Diploidy* (2003b) and A.M.L. Liekens, H.M.M. ten Eikelder, P.A.J. Hilbers, *A Finite Population Model Analysis of Co-Evolution with Matching Pennies* (2003a).

7.1 Introduction

Co-evolution denotes the simultaneous evolution of two or more populations with interdependent or coupled fitness functions. In competitive co-evolution, just like competition in nature, individuals of both populations compete with each other to gather fitness. In cooperative co-evolution, individuals have to cooperate to achieve higher fitness. In genetic algorithms, cooperative co-evolution may be used to decompose solutions for optimization problems into separate parts, and then let co-evolving populations cooperate to find optimal parts of an optimal solution, see e.g., Wiegand (2004).

The interactions among co-evolving populations have previously been modeled in the context of Evolutionary Game Theory (EGT), using replicator dynamics and infinite populations. Similar models have, for example, been used to study equilibriums (Ficici & Pollack 2000b) and comparisons of selection schemes (Ficici, Melnik, & Pollack 2000). Simulations of competitive co-evolution have previously been used to evolve solutions and strategies for small two-player games, e.g., in (Rosin 1997; Lubberts & Miikkulainen 2001), sorting networks (Hillis 1992), or competitive robotics (Floreano, Mondada, & Nolfi 1999).

In this chapter, we provide the construction of a Markov model of co-evolution of two evolutionary systems with finite population sizes. After this construction we compute the relative performances in such a setup, in which a haploid and diploid population co-evolve with each other.

In this chapter, co-evolution is used as a "test bed" to test two populations' relative performance in dynamic environments. Indeed, since the fitness of an individual in one of the co-evolving populations is based on the configuration of the opponent population, the fitness landscapes of both populations constantly change, thereby simulating dynamic environments through both populations' interdependent fitness functions. Note that the results can only be used to discuss the populations' *relative* performance since the dynamics of one population is explicitly dependent on the other.

It is assumed that diploid populations react better to changes in their environment as compared to haploid environments, see Chapter 2. We let haploid and diploid populations face one another in co-evolution, which allows us to study a comparable situation in the history of life on Earth: The first diploid organisms to appear on Earth had to face haploid life forms in a competition for resources. The dynamics of the co-evolutionary competitive games played by these prehistoric cells are similar to the presented models. Correct interpretation of the results may render insights how the earliest diploid life forms were able to compete with haploid life forms.

7.2 Finite Population Models of Co-Evolution

We consider the combined co-evolutionary process of two evolutionary systems, respectively defined by population transitions τ_1 and τ_2, over population search spaces π_1 and π_2. We also use the indexing in the notation to define the corresponding phenotype spaces Ω_1 and Ω_2. We assume that the population sizes are fixed and finite, and their generational transitions are executed at the same rate. In nature, the progress, as measured in generations, may differ from species to species. Especially in co-evolution, the number of generations processed by both co-evolving populations over the same period may differ a lot. In our modeling approach, we assume that all individuals, from both co-evolving populations, simultaneously determine their fitness, and, after that, simultaneously produce their next generations. Other schemes could, for example, evaluate one population and produce a next generation for that population, before evaluating the second population, thereby alternating their generational progress. See Crombach (2002) for an overview of possible schemes.

The fitness functions of both populations are interdependent. The fitness function of an individual in one population is dependent on the composition of the other population. As the evolution in both population progresses, the fitness function for either co-evolving population thus changes over time, as the composition of the populations evolve. If the co-evolution of two populations is competitive, e.g., in a predator-prey setting, an improvement in fitness in one of the populations results in a degradation in the fitness of the other population. In cooperative co-evolution, improvement of fitness in one population may result in an improvement of the fitness in the other population.

We write the fitness function of individuals i in a population P_1, who are co-evolving with a second population P_2 as $f_1(i, P_1, P_2)$, where P_2 now takes the role of the environment $\mathcal{E}$. The fitness of an individual j in population P_2 is $f_2(j, P_2, P_1)$, i.e., population P_1 acts as its environment. As such, the fitness function of an individual in one population is dependent on the configuration of the co-evolving population. Consequently, the transition probabilities of both populations now also depend on the state of the competing population.

The state space π_{co} of the resulting Markov chain of the co-evolutionary system is defined as the Cartesian product of spaces π_1 and π_2, i.e., $\pi_{co} = \pi_1 \times \pi_2$. All (P_1, P_2), with $P_1 \in \pi_1, P_2 \in \pi_2$, are states of the co-evolutionary system. The transition $\tau_{co} : \pi_{co} \to \pi_{co}$ in the co-evolutionary Markov chain of two interdependent Markov chains is defined by

$$\mathbb{P}\left[\tau_{co}((P_1, P_2)) = (P_1', P_2')\right] = \mathbb{P}\left[\tau_1(P_1) = P_1' | P_2\right] \cdot \mathbb{P}\left[\tau_2(P_2) = P_2' | P_1\right] \quad (7.1)$$

where populations P_1 and P_2 are states of π_1 and π_2, respectively. The dependence of τ_1 and τ_2 on P_2 and P_1, respectively, allows for the implementation of a coupled fitness function for either population. Let T denote the $|\pi_{co}| \times |\pi_{co}|$ transition matrix of the co-evolutionary system with transition probabilities

$$T_{(P_1', P_2'),(P_1, P_2)} = \mathbb{P}\left[\tau_{co}((P_1, P_2)) = (P_1', P_2')\right] \quad (7.2)$$

as defined by (7.1).

7.2.1 “Matching Pennies” game as fitness function

In order to construct interdependent fitness functions, we borrow ideas of cooperative and competitive games from Game Theory. In the previous chapter, we have focused on symmetric payoff matrices, where all individuals competing in the games came from the same population. The games adopted for defining the rules in co-evolutionary systems need not to be symmetric. As such, we can play a game where the individuals in the first population have a different goal than the goal imposed on the individuals in the second population.

In the scope of our application, we focus on a family of competitive 2×2 games called “Matching Pennies.” Each of the two players in the game either calls ‘heads’ or ‘tails.’ Consider the payoff matrices for the game in Table 7.1. Depending on the players’ calls and their representative values in the payoff matrices, the players receive a payoff. More specifically, the first player receives payoff $1-K$ if the calls match, and K otherwise. The second player receives 1 minus the first player’s

Q_1	heads	tails
heads	$1-K$	K
tails	K	$1-K$

Q_2	heads	tails
heads	K	$1-K$
tails	$1-K$	K

Table 7.1: Payoff matrices of the Matching Pennies game. One population uses payoff matrix Q_1, where the other players use payoff matrix Q_2. Parameter K denotes the payoff received when the player loses the game, and can range from 0 to 0.5

payoff. We call K the loser's payoff. If K ranges between 0 and 0.5, the first player's goal therefore is to call the same as the second player, whose goal in turn is to do the inverse. Hence the notion of competition in the game. Parameter K can be varied to change the selective pressure in the game. Note that in this case – in contrast with previous occurrences of a selection pressure parameter L – the amount of selective pressure decreases as K increases.

If we assume $0 \leq K < 0.5$, it can easily be seen that there exists a unique Nash equilibrium of this game, where both players call 'heads' or 'tails,' each with probability 0.5. In this equilibrium, both players receive a mean fitness of 0.5. No player can benefit by changing her strategy while the other players keep their strategies unchanged.

Let a population of players denote a finite sized population consisting of individuals who either call 'heads' or 'tails.' In other words, the phenotype spaces of both populations is $\Omega = \{h, t\}$. In our co-evolutionary setup, two models evolving such populations P_1 and P_2 are put against each other. The fitness of individuals in P_1 and P_2 are based on expectations of payoff received when playing the "Matching Pennies" game against random opponents from the other population. We thus assume complete mixing among the individuals to determine the fitness of each individual in either of the populations. Consequently, the fitness of individuals in population P_1 and P_2 are based on payoff matrices Q_1 and Q_2, from Table 7.1, respectively. Let $p_{1,\mathrm{h}}$ denote the proportion of individuals in population P_1 who call 'heads,' and $p_{2,\mathrm{h}}$ the proportion of individuals in P_2 that call 'heads.' Define $p_{1,\mathrm{t}}$ and $p_{2,\mathrm{t}}$ similarly for the proportion of 'tails' in the populations. The fitness of an individual i of population P_1, regarding the constituent strategies of population P_2, can now be defined as

$$f_1(i, P_1, P_2) = \begin{cases} p_{2,\mathrm{h}} \cdot (1-K) + p_{2,\mathrm{t}} \cdot K & \text{if } i \text{ calls 'heads'} \\ p_{2,\mathrm{t}} \cdot (1-K) + p_{2,\mathrm{h}} \cdot K & \text{if } i \text{ calls 'tails'} \end{cases} \tag{7.3}$$

and that of an individual j in population P_2 as

$$f_2(j, P_2, P_1) = \begin{cases} p_{1,\mathrm{h}} \cdot K + p_{1,\mathrm{t}} \cdot (1-K) & \text{if } j \text{ calls 'heads'} \\ p_{1,\mathrm{t}} \cdot K + p_{1,\mathrm{h}} \cdot (1-K) & \text{if } j \text{ calls 'tails'.} \end{cases} \tag{7.4}$$

The mean fitness $\overline{f_1}(P_1, P_2)$ of the individuals in population P_1 is

$$\overline{f_1}(P_1, P_2) = \sum_{i \in \Omega} p_{1,i} f_1(i, P_1, P_2). \tag{7.5}$$

Note that the mean fitness of the individuals in P_2 is closely related to the mean fitness of the individuals in P_1 because of symmetries in the game, as shown by

$$\begin{aligned}
\overline{f_2}(P_2, P_1) &= \sum_{i \in \Omega} f_2(i, P_2, P_1) p_{2,i} \\
&= f_2(h, P_2, P_1) p_{2,h} + f_2(t, P_2, P_1) p_{2,t} \\
&= (p_{1,h} K + p_{1,t}(1 - K)) p_{2,h} + (p_{1,t} K + p_{1,h}(1 - K)) p_{2,t} \\
&= (p_{2,h} K + p_{2,t}(1 - K)) p_{1,h} + (p_{2,t} K + p_{2,h}(1 - K)) p_{1,t} \\
&= (p_{2,h} K + p_{2,t}(1 - K))(1 - p_{1,t}) + (p_{2,t} K + p_{2,h}(1 - K))(1 - p_{1,h}) \\
&= p_{2,h} K + p_{2,t}(1 - K) + p_{2,t} K + p_{2,h}(1 - K) - \overline{f_1}(P_1, P_2) \\
&= p_{2,h} + p_{2,t} - \overline{f_1}(P_1, P_2).
\end{aligned}$$

Hence, we have shown that

$$\overline{f_2}(P_2, P_1) = 1 - \overline{f_1}(P_1, P_2) \tag{7.6}$$

always holds for the Mathcing Pennies game.

7.2.2 Limit behavior

In Section B.2.2 of the appendix, we show that the combination of irreducible and aperiodic interdependent Markov chains, as defined above, does not generally result in an irreducible and aperiodic Markov chain. Therefore, we cannot simply assume that the Markov chain that defines the co-evolutionary process converges to a unique fixed point.

We can, however, make the following assumptions: If mutation can map any individual – in both of the co-evolving models – to any other individual in the genotype space with a strictly positive probability, then all elements in the transition matrices of both co-evolving Markov chains are always nonzero and strictly positive, see Theorem 3.2. As a result from multiplying the transition probabilities in (7.1), all transition probabilities of the co-evolutionary Markov chain are then also strictly positive. This makes the combined Markov chain irreducible and aperiodic, such that the limit behavior of the whole co-evolutionary process can be studied by finding the unique eigenvector, with corresponding eigenvalue 1, of the transition matrix as defined by (7.1), due to the Perron-Frobenius theorem, see Appendix B.

7.2.3 Expected performance

According to the definition of the co-evolutionary system in (7.1), the transition matrix for a given set of parameters can be computed. The unique stochastic eigenvector, with corresponding eigenvalue 1, of this transition matrix can be found using the power method, through iterated multiplication of the transition matrix with an initially distributed stochastic vector. We then adopt the resulting eigenvector to measure the expected relative long term performances of the co-evolutionary model.

Assuming strictly positive rates of mutation, let $\mathbf{x}^*$ denote the unique stochastic eigenvector, with corresponding eigenvalue 1, of transition matrix T of the co-evolutionary system. Vector $\mathbf{x}^*$ denotes the fixed point distribution of states of the co-evolutionary system, with component $x^*_{(P_1,P_2)}$ denoting the probability of ending up in state $(P_1, P_2) \in \pi_{co}$ in the limit. If $\overline{f_1}(P_1, P_2)$ gives the mean or expected fitness of the individuals in population P_1, given an opponent population P_2, then

$$\mathbb{E}\left[f_1\right] = \sum_{(P_1,P_2)\in\pi_{co}} x^*_{(P_1,P_2)} \cdot \overline{f_1}(P_1, P_2) \tag{7.7}$$

gives the expected fitness in the long run of the dynamics of the first population in the limit, in relation to its co-evolving population. Similarly, the expected long run fitness of the second population can be computed. Because of (7.6) we know that the expected fitness of one population is always one minus the expected fitness of the other population, since

$$\begin{aligned}
\mathbb{E}\left[f_2\right] &= \sum_{(P_2,P_1)\in\pi_{co}} x^*_{(P_1,P_2)} \cdot \overline{f_2}(P_2, P_1) \\
&= \sum_{(P_2,P_1)\in\pi_{co}} x^*_{(P_1,P_2)} \cdot (1 - \overline{f_1}(P_1, P_2)) \\
&= \sum_{(P_2,P_1)\in\pi_{co}} x^*_{(P_1,P_2)} - \sum_{(P_2,P_1)\in\pi_{co}} x^*_{(P_1,P_2)} \cdot \overline{f_1}(P_1, P_2),
\end{aligned}$$

hence,

$$\mathbb{E}\left[f_2\right] = 1 - \mathbb{E}\left[f_1\right]. \tag{7.8}$$

We can also compute the variance and standard deviation in order to discuss the significance of the exact results.

Given the parameters for fitness determination, selection and reproduction of both co-evolving populations in the co-evolutionary system, we can now estimate the mean fitness, and discuss the performance of both evolutionary systems, in the context of their respective competitors' performance.

We adopt the expected fitness of a population in the limit as its measure of performance, in relation to its opponent, co-evolving population. We have observed earlier on that playing either heads or tails with a probability 0.5 is the Nash equilibrium strategy of this game. Similarly, the evolutionarily stable strategy (ESS) of the game is to play either strategy with 50% probability, which gives all individuals a fitness of 0.5. Taking this again further to the context of expected performances, it is expected that equally capable evolutionary systems have an expected performance, or expected fitness of 0.5. Any deviation from this – because of (7.8) an increase in one population's expected performance denotes an equally large decrease in performance for the other population – indicates that one population is expected to perform better at its co-evolutionary task than its opponent population.

7.3 Results

7.3.1 Haploid versus haploid

Mutation rate. Figure 7.1 shows the expected performance and standard deviation of the system evolving population P_1, with $K = 0$, fixed population sizes $r_{P_1} = r_{P_2} = 10$ and variable mutation rates μ_{P_1} and μ_{P_2}. If μ_{P_1} equals μ_{P_2}, then the expected mean fitness of both populations is 0.5, as predicted.

It turns out that, under the given parameter settings, the system with higher mutation rate performs better in the long run. Indeed, with smaller mutation rates, populations get more easily fixated in a population, where the other population with a higher mutation rate is more able to change the current prominent strategy. For small mutation rates, the standard deviation is high, since populations may become more easily fixated in either population that consist of a high proportion of either strategy. Indeed, in the extreme where the mutation rate of the population is 0, the population is either in the population with all 0s or all 1s. As both populations have a sufficiently small mutation rate, the system becomes repeatedly fixated in the situation where either population is fixated, consequently resulting in a high standard deviation.

Again, the observation that higher mutation rates result in better performance differs from the observation in static fitness functions, where small mutation rates are preferred for better convergence, or exploitation of the optimal genotype. In the dynamics of two co-evolving populations involved in the Matching Pennies game, the populations must be able to switch quickly from high proportions of one strategy, to high proportions of the other.

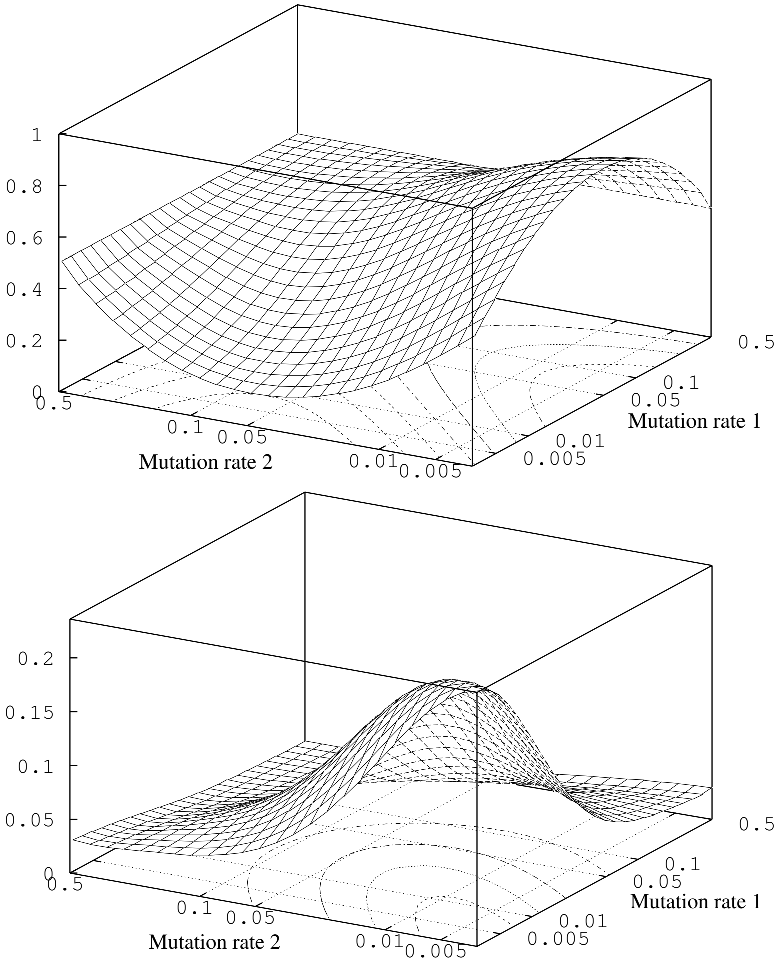


Figure 7.1: Expected performance (top) and standard deviation about this mean (bottom) of population P_1 for various settings of mutation rates μ_{P_1} and μ_{P_2}. The population sizes of both systems are fixed to 10 individuals. Parameter K has been set to 0

If the strategy adopted by the first population matches that of the second, this latter population needs to change its strategy to gather more payoff. As a result of such a change by the second population, the strategies adopted in the populations differ, giving the second population a benefit in payoff over the first. Now, the first population has, again, to try and match the calls of the players in the second population, restarting the dynamical cycle in the Matching Pennies game. As a consequence, since the fitness of a strategy is dependent on the dynamics of the competing strategy, and as it needs to adapt to the beneficial strategy dependent on the strategy of the opponent population, it is required to be more agile than the opponent strategy in order to acquire high payoffs, and a high performance. A higher mutation rate offers such agility, and therefore allows the population with a higher mutation rate an advantage, and thus a higher expected performance.

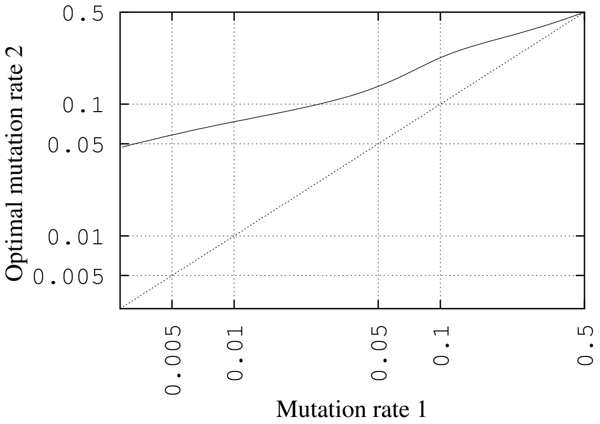


Figure 7.2: Optimal setting for μ_{P_1} if μ_{P_2} is given, such that population P_1 performs best

If μ_{P_2} is set and we are given the task to choose μ_{P_1} such that the performance of P_1 is best, or vice versa, then Figure 7.2 gives the optimal mutation rate μ_{P_1} for the first population in terms of the mutation rate μ_{P_2} of the second.

Selective pressure. So far, we only considered a fixed payoff $K = 0$ for the experiments. Using this setting of the "Matching Pennies" game puts a high selective pressure on the populations in the co-evolutionary system.

In Figure 7.3, the performance and standard deviation of P_1 is depicted for various settings of μ_{P_1} and K. For any setting of K, the performance of P_1 is higher if its mutation rate μ_{P_1} is higher than $\mu_{P_2} = 0.05$, which thus extends our previous observation that a higher mutation rate than the competing population provides a higher expected performance. As K comes closer to 0.5, the differences in performance become less significant.

Population size. Figure 7.4 shows the influence of population sizes r_{P_1} and r_{P_2} on the performance of population P_1. Mutation rates μ_{P_1} and μ_{P_2} have been set to 0.1, and parameter K equals 0. As r_{P_1} equals r_{P_2}, the performance of both populations is 0.5. In all other cases, the system with the larger population performs better than its opponent population, although this difference in performance is only significant as the ratio of population sizes r_{P_1}/r_{P_2} strongly differs from 1.

In similarity with the previous models, the standard deviation about the expectations is highest for small population sizes. Again, this is due to the fact that systems with small population sizes become more easily fixated in populations with a high proportion of either strategy. These effects of random genetic drift are most common in populations with a small mutation rate and small population size.

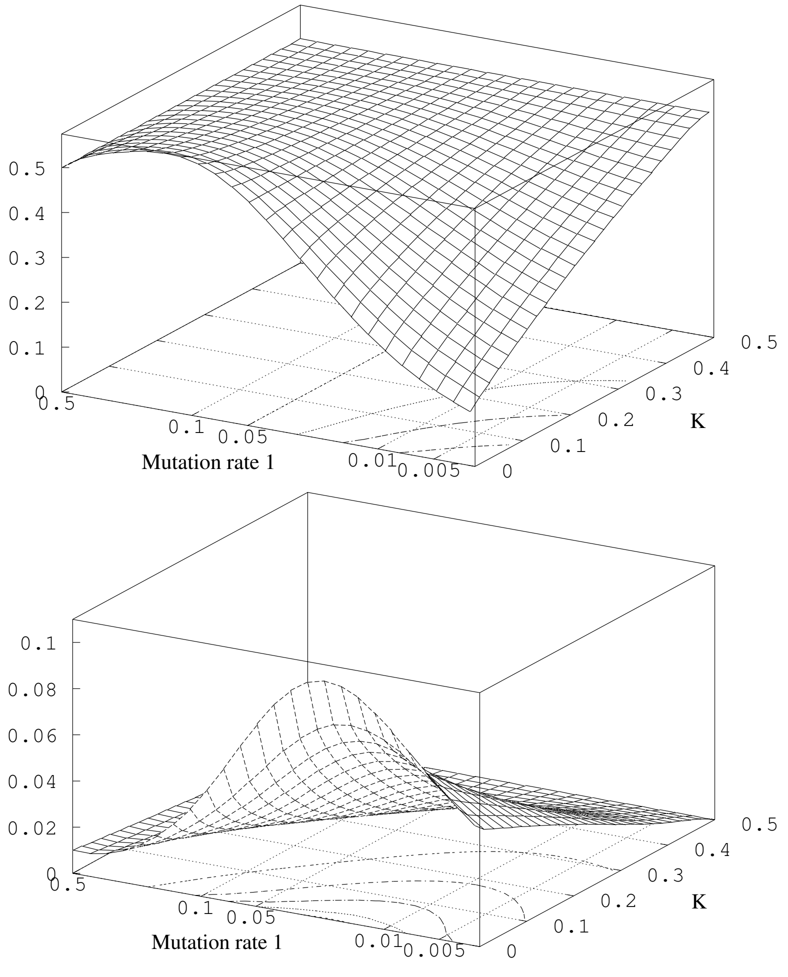


Figure 7.3: Expected performance (top) and standard deviation (bottom) of population P_1 for various settings of mutation rate μ_{P_1} and variable payoff parameter K. Population sizes r_{P_1} and r_{P_2} have been fixed to 10. The mutation rate of P_2, μ_{P_2} is set to 0.05

7.3.2 Haploid versus diploid

For the Matching Pennies game, we construct a co-evolutionary Markov chain in which a haploid and diploid population compete with each other. With this construction, and their transition matrices, we can determine the performance of both populations according to the limit behavior of the Markov chain.

Let π_{co} be the search space of the co-evolutionary system, defined by the Cartesian product of the haploid populations' search space π_H and diploid populations π_D, such that $\pi_{co} = \pi_H \times \pi_D$. Depending on a fixed population size r for both competing populations, $|\pi_{co}| = ((r+2)(r+1)^2)/2$ denotes the size of the co-evolutionary state space, based on the size of the population spaces of both models as in (3.3).

For any state $(P_1, P_2) \in \pi_{co}$, with P_1 a haploid and P_2 a diploid population,

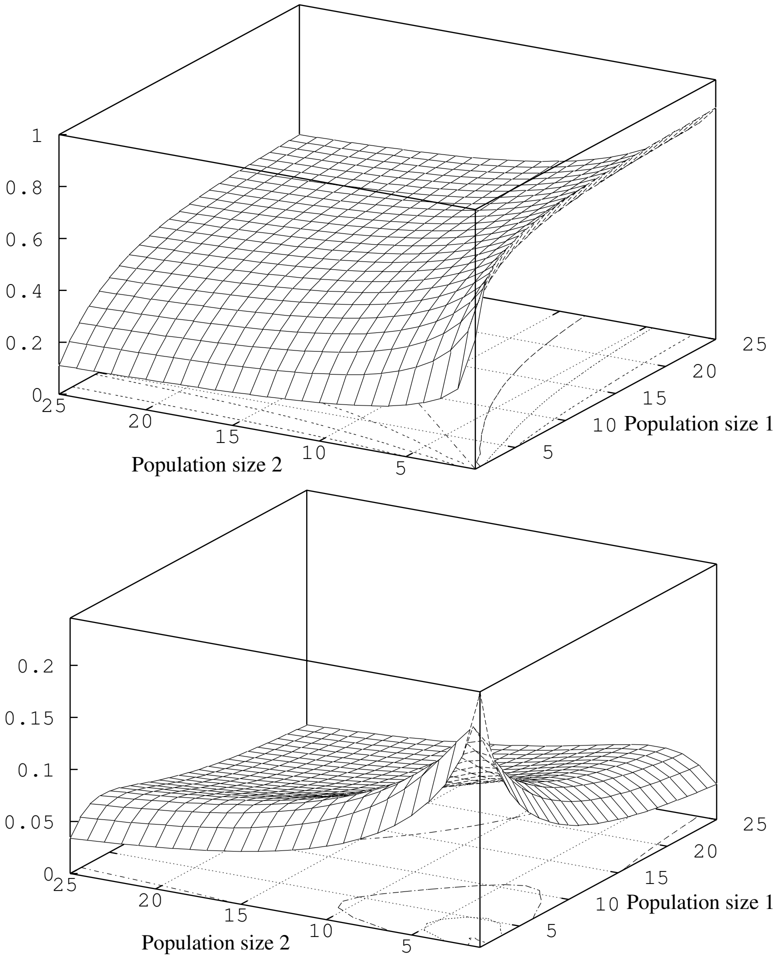


Figure 7.4: Expected performance (top) and standard deviation (bottom) of population P_1 for various settings of population sizes r_{P_1} and r_{P_2}

let (7.3) and (7.4) be the respective fitness functions for the individuals in the haploid and diploid populations. Since we want to compare the populations' performance under comparable conditions, both populations are assumed to have the same parameters for their genetic operators.

Pure dominance. Let 1 be the dominant allele, and 0 the recessive allele in diploid heterozygous individuals. This implies that diploid individuals with genotype $\{0, 1\}$ have phenotype 1. If we would choose 0 as the dominant allele instead of 1, the co-evolutionary system would yield the exact same performance results, because of symmetries in the Matching Pennies game. The same holds for exchanging fitness functions f_1 and f_2.

Figure 7.5(a,b) shows the expected fitness of the haploid population in the long run. For large mutation rates, the haploid system generally performs sig-

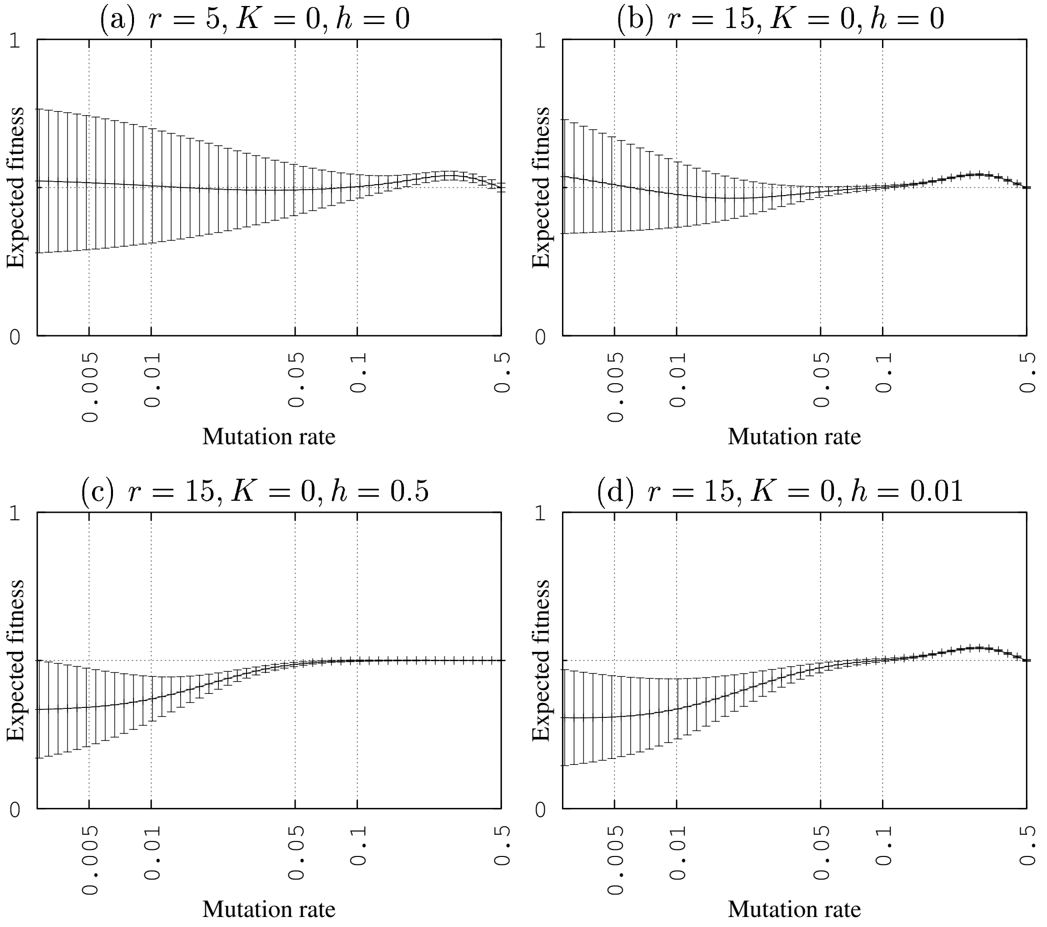


Figure 7.5: Expected performance and standard deviation of the haploid population in the co-evolutionary system, for variable mutation rates μ and various dominance coefficients. The mean fitness of the diploid population always equals 1 minus the mean fitness of the haploid population. The size of both populations is fixed to 5 in (a) and 15 in (b-d).

nificantly better than the diploid model. The small set of parameter settings for which diploidy performs better, increases as the population size is increased. The difference in performance, however, is not significant for the performed measurements with pure dominance since the standard deviation about the expectation is too large.

Partial dominance. Instead of using a pure dominance scheme in the diploid model, we can also assign a partial dominance scheme to the dominance operator. In this dominance scheme with coefficient of dominance h, the heterozygous genotype $\{0, 1\}$ has phenotype 0 with probability h, and phenotype 1 with probability $1 - h$. The dominance degree is the probability that the recessive allele is expressed in the case of heterozygosity. Since our model is stochastic, we could also state that the fitness of an heterozygous individual is an intermediate of the fitness of both homozygous phenotypes.

Expected performance of the haploid population for various dominance coefficients are depicted in Figure 7.5(c,d). Even for small coefficients of dominance, significantly better performance results for the diploid population can be observed for small mutation rates and high selection pressure (small K), in relation to the haploid population. Even a small coefficient of dominance allows the recessive alleles to be expressed in the case of heterozygosity. This allows the diploid population to test either strategy more often than a similar haploid population. Because of the partial dominance, the diploid population appears as having a larger population size, which gives the diploid model more agility to adapt to changes in the haploid population, working in its advantage. For sufficiently large mutation rates (approximately $\mu > 0.1$) and low coefficients of dominance, the haploid population still performs significantly better as compared to the diploid system.

Summary

We have constructed a Markov model of finite populations involved in co-evolution. The model allows us to study the relative performance of two co-evolving populations in the long run. We have adopted the Matching Pennies game for the interdependent fitness function, and studied the performance of two co-evolving populations with this game. We have shown that having a higher mutation rate than your opponent renders higher expected payoff in relation to that opponent. This observation is independent of the selective pressure. Larger populations render higher expectations of fitness, although the ratio of population sizes should differ strongly from 1 in order to be significant.

The application of co-evolving a haploid versus a diploid population shows that, given the Matching Pennies game, and if pure dominance is assumed, the results are only in favor of diploidy in case of specific parameter settings. Even then, the results are not significant and subject to a large standard deviation. A diploid population with partial dominance and a strictly positive dominance degree can significantly outperform a haploid population, if similar conditions hold for both populations. These results are expressed best under low mutation pressure and high selection pressure. Diploidy performs relatively better as the population size increases.

Chapter 8

Conclusions and Discussion

The size of an evolving population of individuals is an important parameter of the population's evolutionary dynamics. In very large populations a vast number of gene combinations and variations can be tested. Even if deleterious mutants are introduced into the population, they most probably die out because of the substantial number of selectively advantageous individuals. The effects of random events in a large population are negligible.

However, stochastic effects are of great importance in small populations. The probability that an *unlucky* sample of alleles is inherited by a child population is higher if the offspring population is smaller. It is a common fact from statistics that small samples render unrepresentative descriptions. Similarly, the proportions of alleles in a small child population are usually not representative for the composition of alleles in the parent population.

During the evolution of a small population, such unlucky samples can build up generation after generation, and allow the composition of the population to genetically drift away from the initial configuration. Eventually, some alleles are lost from the finite population, while others become fixated. It is not necessarily true that the alleles adopted by the small population are selectively advantageous. Deleterious alleles can more easily take over a population as it has less individuals. Slow mutation rates and low selective pressure on deleterious alleles further amplify this effect. A smaller population suffers more from loss of genetic diversity, and the evolvability of a population is lost as its size is smaller. As a result, a sufficiently small population may be unable to adapt to its environment. The population's environment may change over time, forcing the group to continuously adapt and keep track of these changes. Because of the loss of evolvability, a small population may become unable to cope with the speed of environmental changes.

In this thesis, we have constructed mathematical models to study the evolution of small populations in such dynamically changing environments. We have built Markov models of finite evolving populations of abstract individuals. Markov models allow us to compute the limit behavior of the system, which can then be interpreted as the expected behavior of the population after a large number of generations. The predictions of a Markov model are exact with respect to the numerical implementation of the power method. The influence of different parameters – such as the population size, the rate of mutation, ploidy of the individuals, the amount of selective pressure, or the speed of changes in the environment – on the long run behavior of the models can be studied. By comparing the expectations of such finite models with the predictions of models that assume infinitely large populations, and by comparing models with differing population size, we can gain insights in what effect the population size has on the evolutionary behavior of a small population in a dynamic environment.

We have constructed models of finite populations with three types of dynamic environments to study these effects. In the first model we have constructed dynamic environments by implying environmental changes at given time intervals. As such, we can control the speed of changes in a dynamic environment and study the impact on the evolutionary dynamics of small populations. In the second model we have applied techniques from evolutionary game theory to model dynamic environments. In these models, an individual's fitness is dependent on other members of its population. As the population changes over time, the fitness of each individual changes as a result. The evolutionary dynamics of the population itself thus governs the dynamics of the environment. In the last model, we have constructed models of co-evolution of interdependent populations. As one population evolves, the environment of the individuals in the other co-evolving populations changes.

Because of the modeling approach with Markov models, which requires the enumeration of all possible states of the system, the populations and their environments studied in this thesis are limited. Genotype spaces and population sizes had to be kept relatively small to allow for the numerical computation of the systems' limit behavior. The models studied in this thesis are however considerably large. Because of our parallel implementation of the power method, we have been able to study state spaces surpassing previously established records in the study of evolutionary systems. For small genotype spaces, we were able to predict the limit behavior of relatively large populations. As a result, we have a good idea of how the behavior of simple games and simple genotype spaces scale up with the population size. In attempts to study small populations of individuals in genotype spaces that combine multiple genes, population sizes have been very limited. How such models scale with increasing population sizes and increasing number of genes remains inconclusive.

Since the focus of this thesis is on dynamic environments, no complex genotype spaces or genotype-phenotype mappings have been considered, as they would have only clouded our view of the evolutionary dynamics under study. We however acknowledge that non-linear genotype-phenotype mappings may have an important influence on the dynamics of the population in its dynamic environment. As future work on this topic, a specific methodology other than Markov models would be required – e.g., through the usage of a statistical mechanics approach or lossy compression of the transition-probability matrices – which are not exact but render good approximations of more complex genotype spaces and larger population sizes. The adopted methodologies in this thesis for defining specific dynamic environments are however universal and can be adapted upon by other methods.

In static environments, finite populations are commonly known to become fixated in alleles. If the selective pressure exerted on the deleterious alleles is large enough, and a small, but nonzero rate of mutations is available, a sufficiently large population contains a high proportion of advantageous alleles with a high probability in the long run. For very small populations, the stochastic effects of genetic drift can dominate the behavior of the population, as they become stronger than the forces of natural selection as the population becomes smaller.

In the setting where the population itself, or co-evolving populations determine the dynamics of the environment, an infinite population may converge to a stable situation in the limit. In evolutionary game theory, the population is said to adopt an evolutionarily stable strategy, i.e., a strategy that cannot be invaded by a mutant strategy. Since the dynamics converge in the limit, the long run expectations of a finite population are similar to those of a static fitness environment. For finite populations that are sufficiently large, the expected long run behavior of the system is similar to the infinite model. If a mutant strategy is introduced in a finite population of individuals adopting an evolutionarily stable strategy, there is a small probability that an unlucky sample allows the deleterious mutant to take over the population. Because of stochastic elements, such as mutations and a small population size, the system may become biased toward a strategy that is not evolutionarily stable, and thus contradicts the expectations of the infinite population model. With small populations, the concept of an evolutionarily stable strategy becomes obsolete. For specific games, the parameter settings of the population size and of the mutation rate can have a significant effect on the long run behavior of the system.

In dynamic environments that perpetually undergo considerable changes, the effects of random sampling and genetic drift become more apparent in the long term behavior of an evolutionary system. During periods where the dynamic environment remains relatively static, a small population may be able to discover advantageous alleles and has a chance of becoming fixated on these alleles. As the period of stasis is longer, the probability of discovering the advantageous alleles is higher. But as the environment changes dramatically after such a static period, a fixated finite population does not show the necessary diversity to quickly adapt to the new environment. A dynamic environment may repetitively change the advantage of one allele over the others. At each environmental distortion, the finite population needs to escape from its state of fixation to discover the advantageous alleles of the new environment. The resulting process of repetitive fixation and continuing environmental pressure to escape from the state of fixation slows down the evolutionary progress of a small population in its dynamic environment. Since smaller populations are more affected by effects of genetic drift and fixation, they also suffer more from this slowing effect, and consequently perform less well than larger populations. An infinitely large population is least affected by this slowing effect since it does not become repeatedly fixated in alleles during periods of environmental stasis. An infinite population generally does not become fixated, and demonstrates an unbounded amount of diversity, allowing it to more quickly adapt to the changes in the environment.

A finite population needs a sufficiently high mutation rate to adapt to environmental changes. Mutations induce higher diversity, and thus provide a way to escape from the repetitive states of fixation. The mutation rate, however, must not be too high such that the population can exploit newly discovered advantageous alleles within the period of environmental stasis. In order to perform best in a given dynamic environment, a mutation rate is thus required such that it balances the exploration and exploitation of advantageous alleles. As the environment changes more frequently, the rate of mutations must be higher to perform best. As the size of the population is smaller, the mutation rate must also be higher since the probability of getting fixated in deleterious alleles is higher.

With models that involve the co-evolution of multiple populations, we have studied the performance of two competing populations. In competitive co-evolution, one population's alleles become deleterious as the opponent population discovers advantageous alleles. The performance of one population is expressed with respect to the evolutionary dynamics of the opponent population. If one population has a higher mutation rate than its opponent, then this former population is able to perform better than its competitor. Indeed, the population with a higher mutation rate is more diverse and more flexible at adapting its strategy according to the evolutionary dynamics of the opponent population. If one of either populations is larger, then this population has also the advantage of being less affected by genetic drift and fixation, allowing it to perform better than the smaller population. The difference in population size, however, must be sufficiently large for the difference in performance to be significant.

It is commonly hypothesized that populations of diploid individuals are better equipped for coping with environmental changes than populations of haploid individuals. Firstly, a diploid population of equivalent size demonstrates increased diversity, which helps overcoming the problems of fixation. Secondly, the process of dominance, which exists in diploid individuals and not in haploid individuals, allow an implicit memory of advantageous solutions to previously encountered environmental situations to develop. Most of the observations in this text, however, do not show a significant advantage of diploid individuals in dynamic environments over haploid individuals. Only in the case of a high degree of dominance, a significant advantage of adopting diploidy was observed. It should however be noted that the level of abstraction in our models may be too narrow for the benefits of diploidy to come into effect. Possible adaptations to the current model can include more complex dominance schemes, possibly with dominance of alleles embedded into the genotype of the individual, or more complex genotype spaces than the ones considered in this thesis. We, however, remain cautious about the hypothesis that a population of diploid individuals is indeed better suited for dynamically changing fitness environments. The study of larger and more complex genotype spaces may offer a more accurate analysis of the hypothesis.

The observation that small populations repetitively become fixated in dynamic environments is analogous to the observation that finite populations induce states of metastability in static environments. In a static environment, finite populations become temporarily fixated in genotypes, before moving on to explore genotypes with higher fitness. These states of temporarily stability, intermitted by bursts of rapid evolutionary progress are known as punctuated equilibria. In finite populations, punctuated equilibria are more frequently observed than in infinite populations, due to the increased probability of fixation in finite populations. Whereas a population is expected to find an optimal combination of alleles for a static environment in the long run, the perpetually repetitive changes in a dynamic environment actively invoke such punctuated equilibria, even in the long run.

A clear example of this behavior was observed in the evolutionary dynamics of a finite population playing Rock-Paper-Scissors. Repeatedly, the population becomes fixated in one strategy, until its dominating strategy suddenly takes over the population in a rapid burst of evolutionary innovation. As with static environments, a finite population size encourages this behavior, and forces the finite population to evolve more slowly from one strategy to the next. In contrast with the common study of complex genotype spaces to bring forth punctuated equilibria, simple games with cyclic behavior offer a modeling approach that is mathematically more accessible. We suggest that evolutionary games with cyclic behavior may consequently be considered for the future study of punctuated equilibria.

Appendix A

Enumerating Populations

This appendix details the enumeration of population vectors and conversion to and from their associated numbers. This computation is required when Markov models with a very large number of possible populations are implemented, and is reported on by van Heeswijk (2004).

Population vectors. For a finite population P of size r, let $P_i = rp_i$ denote the number of individuals of type $i \in \Omega$ in this population with $\sum_{i \in \Omega} P_i = r$. A population can thus be written as a population vector $\mathbf{P} = r\mathbf{p} = (P_0, P_1, \ldots, P_{n-1})^\top$.

As an example, assume a population of $r = 3$ individuals over a search space of $n = 4$ types of individuals, i.e., Ω corresponds to $\{0, 1, 2, 3\}$. Consequently, there are $\left(\begin{array}{c} r+n-1 \\ r \end{array} \right) = \left(\begin{array}{c} 3+4-1 \\ 3 \end{array} \right) = 20$ possible populations, see Equation 3.3. Table A.1 gives a sorted list of all possible population vectors for these parameters.

#**P**	(P_0,	P_1,	P_2,	P_3)	#**P**	(P_0,	P_1,	P_2,	P_3)
0	(0,	0,	0,	3)	10	(1,	0,	0,	2)
1	(0,	0,	1,	2)	11	(1,	0,	1,	1)
2	(0,	0,	2,	1)	12	(1,	0,	2,	0)
3	(0,	0,	3,	0)	13	(1,	1,	0,	1)
4	(0,	1,	0,	2)	14	(1,	1,	1,	0)
5	(0,	1,	1,	1)	15	(1,	2,	0,	0)
6	(0,	1,	2,	0)	16	(2,	0,	0,	1)
7	(0,	2,	0,	1)	17	(2,	0,	1,	0)
8	(0,	2,	1,	0)	18	(2,	1,	0,	0)
9	(0,	3,	0,	0)	19	(3,	0,	0,	0)

Table A.1: Sorted list of (transposed) population vectors for $r = 3$ and $n = 4$

The population vectors in this table have been sorted increasingly, and numbered $0, \ldots, |\pi| - 1$. Iterating over all possible populations to determine a population vector's population number, or vice versa, to determine the composition of a population for a given population number is a computationally expensive task, and would require $O\left((r+n)^r\right)$ operations, see Equation (3.4). Note that the computational complexity of this computation is expressed in both r and n, as both parameters may grow very large, and both may have to be taken into account. Population size r can grow large, as large populations are studied. The size n of the search space may also grow very large, e.g., most optimization problems have very large search spaces.

We use a direct, computationally less expensive method to do conversions among these two.

Determining the population number of a population vector. We can determine the number $\#\mathbf{P}$ of a population vector $\mathbf{P}$ recursively. As a basis for this recursion, we can easily determine the number of a population vector that is defined over only 2 individual types, i.e.,

$$\#(P_0, P_1)^\top = P_0.$$

In the induction step of our recursion, we determine the number of a population vector with one more individual type, with

$$\#(P_0, P_1, \ldots, P_{n-1})^\top = \sum_{i=0}^{P_0-1} \binom{r_P - i + n - 2}{n-2} + \#(P_1, P_2, \ldots, P_{n-1})^\top \quad \text{(A.1)}$$

where $r_P = \sum_{i=0}^{n-1} P_i$. The sum of combinations in this formula gives the number of populations whose first element in their population vector is smaller than P_0. Note that $\#(0, P_1, P_2, \ldots, P_{n-1})^\top = \#(P_1, P_2, \ldots, P_{n-1})^\top$. In order to compute a population vector's number, we do not need to know the search space's size or the population's since these are implicitly defined by the vector's dimension (n) and elements ($r = \sum_{i=0}^{n-1} P_i$).

The algorithm requires $O(r)$ computations of a factorial to determine the population number of a population vector. Indeed, at the ith iteration of the algorithm, P_i factorials are computed, which gives a total of $P_0 + \cdots + P_{n-1} = r$ computations of a factorial for the complete algorithm.

Example A.1 (Computing $\#(2, 0, 1, 0)^\top$). As an example, consider population vector $(2, 0, 1, 0)^\top$, with $r = 3$ and $n = 4$. Its corresponding number is computed with

$$\begin{aligned}
\#(2, 0, 1, 0)^\top &= \binom{3 - 0 + 4 - 2}{4 - 2} + \binom{3 - 1 + 4 - 2}{4 - 2} + \#(0, 1, 0)^\top \\
&= 10 + 6 + \#(1, 0)^\top \qquad \text{(A.2)} \\
&= 16 + 1 \\
&= 17.
\end{aligned}$$

Table A.1 verifies this result. The intermediate results 10 and 6 in step (A.2) denote the number of populations $\mathbf{P}$ (of size $r = 3$ with $n = 4$) with $P_0 = 0$ and $P_0 = 1$ respectively. □

Extracting the population vector from a population number. Going the other way, we also want to determine population vector $\mathbf{P} = (P_0, P_1, \ldots, P_{n-1})^\top$ given its population number $\#\mathbf{P}$, population size r and search space size n. The first element of this vector can easily be computed with

$$P_0 = \max\left\{ k \mid \sum_{i=0}^{k-1} \left(\begin{array}{c} r-i+n-2 \\ n-2 \end{array} \right) < \#\mathbf{P} \right\}.$$

Similar to (A.1), the sum of combinations computes the number of populations whose first element in their population vector is smaller than P_0. All of these vectors have a population number smaller than $\#\mathbf{P}$.

We can determine the following elements of $\mathbf{P}$ using recursion of this equation. Element P_1 of our vector is equal to the first element P_0' of population vector $\mathbf{P}'$ with population number

$$\#\mathbf{P}' = \#\mathbf{P} - \sum_{i=0}^{P_0} \left(\begin{array}{c} r-i+n-2 \\ n-2 \end{array} \right),$$

with population size $r' = r - P_0$ and search space size $n' = n - 1$.

As the basis for this recursion, we know that a population number $\#\mathbf{P}$ of size r over 2 individual types corresponds with population vector $(P_0, P_1)^\top$ where $P_0 = \#\mathbf{P}$ and $P_1 = r - \#\mathbf{P}$.

Similar to the first algorithm, the algorithm for extracting a population vector from a population requires $O(n + r)$ computations of a factorial. Indeed, at each ith iteration we compute one too many factorial to find the maximum, i.e., $P_i + 1$ factorials at each iteration, giving a total of $P_0 + 1 + P_1 + 1 + \cdots + P_{n_1} + 1 = r + n$ factorials for the whole algorithm.

Example A.2 (Extracting P from $\#\mathbf{P} = 12, r = 3, n = 4$). As an example, consider a population of $r = 3$ individuals over a search space of $n = 4$ types of individuals. We want to determine the population vector $\mathbf{P}$ that corresponds to population number $\#\mathbf{P} = 12$.

As a first step, we determine P_0 with

$$P_0 = \max\left\{ k \mid \sum_{i=0}^{k-1} \left(\begin{array}{c} 5-i \\ 2 \end{array} \right) < 12 \right\}.$$

For $k = 0, 1, 2$, $\sum_{i=0}^{k-1} \left(\begin{array}{c} 5-i \\ 2 \end{array} \right)$ equals $0, 10, 16$ respectively. The maximum k for which the in equation holds gives $P_0 = 1$.

Secondly, we can find P_1 with $P_1 = P_0'$ being the first element of the population vector of population $\mathbf{P}'$ with population size $r' = r - P_0 = 2$, over search space Ω'

with size $n' = n - 1 = 3$, and whose number is $\#\mathbf{P}' = \#\mathbf{P} - 10 = 2$. Repeating our first step with this smaller population vector yields

$$P_1 = P_0' = \max\left\{ k \mid \sum_{i=0}^{k-1} \binom{3-i}{2} < 2 \right\} = 0.$$

As a third iteration step, we need to compute the elements of population vector $\mathbf{P}'' = (P_0'', P_1'')^\top = (P_2, P_3)^\top$ with $\#\mathbf{P}'' = 2$, $r'' = 2$ and $n'' = 2$. From our recursion basis, we can compute these elements easily with $P_2 = P_0'' = \#\mathbf{P}'' = 2$ and $P_3 = P_1'' = r'' - \#\mathbf{P}'' = 0$.

The resulting population vector now equals $\mathbf{P} = (1, 0, 2, 0)^\top$, which can be verified in Table A.1 as the population vector with corresponding population number $\#\mathbf{P} = 12$. □

Appendix B

Coupling of Finite Markov Models

Some models in this thesis use combinations, or products, of Markov models. These models – along with their limit behaviors – are then studied by finding the unique eigenvector, with corresponding eigenvalue 1, of the combined Markov model. In order to supply a mathematical basis for these combined models, and to study the existence of their limit distributions, this chapter discusses the ergodicity of coupled Markov models.

B.1 Properties of Markov Models

Before we can introduce the notion of the combination or product of Markov models, we give some properties of irreducible and aperiodic Markov models.

B.1.1 Finite, discrete time Markov models

Definition B.1 (Markov model). A Markov model is a tuple $\langle \pi, \tau \rangle$ with state space π and probabilistic transition function $\tau : \pi \to \pi$.

For each state $i \in \pi$, $\tau(i)$ hence gives a distribution over all states in π.

Definition B.2 (Finite Markov model). Markov model $\langle \pi, \tau \rangle$ is said to be finite if $|\pi|$ is finite.

In this thesis, we assume that the state spaces used for the Markov models are finite, hence we assume finite Markov models for the rest of this chapter.

The transition matrix T of Markov model $\langle \pi, \tau \rangle$ is an $|\pi| \times |\pi|$ matrix that contains the transition probabilities $P_{ij} = \mathbb{P}\left[\tau(j) = i\right]$ for all $i, j \in \pi$. We say that state i can be reached in one step from state j if $T_{ij} > 0$.

Definition B.3 (Nonnegative vector). A vector $\mathbf{x}$ is said to be nonnegative if all vector elements are nonnegative.

Definition B.4 (Stochastic vector). A vector $\mathbf{x}$ with n elements is said to be stochastic if the vector is nonnegative and the sum of its elements equals 1, i.e., $\sum_{i=1}^{n} x_i = 1$.

Definition B.5 (Stochastic matrix). An $n \times n$ matrix T is said to be stochastic if its column vectors are stochastic.

A transition matrix of a Markov model is a stochastic matrix.

Theorem B.1. *If $n \times n$ matrix T and vector $\mathbf{x}$ with n elements are stochastic, then $T \cdot \mathbf{x}$ is stochastic.*

Proof. Due to multiplication and summation of nonnegative elements in matrix T and vector $\mathbf{x}$, all elements in $T \cdot \mathbf{x}$ are nonnegative.

The sum of elements in vector $T \cdot \mathbf{x}$ is equal to

$$\sum_{i=1}^{n} (T \cdot \mathbf{x})_i = \sum_{i=1}^{n} \sum_{j=1}^{n} T_{ij} x_j = \sum_{j=1}^{n} x_j \cdot \sum_{i=1}^{n} T_{ij} = \sum_{j=1}^{n} x_j \cdot 1 = 1.$$

Hence, $T \cdot \mathbf{x}$ is stochastic. □

Assume a Markov model $\langle \pi, \tau \rangle$ with stochastic transition matrix T and the stochastic vector $\mathbf{x}_t$ with size $|\pi|$ denoting the distribution over state space π at a given time t. Stochastic vector $\mathbf{x}_{t+1} = T \cdot \mathbf{x}_t$ now gives the distribution at time $t + 1$. This equation is known as the Chapman-Kolmogorov equation, see e.g., Papoulis & Unnikrishna (2002). Since we assume models with discrete time steps, the Markov models we focus on here are discrete time Markov models (DTMM).

B.1.2 Properties

We go on with the definitions of irreducibility and aperiodicity, which are important and necessary properties of Markov models as we become interested in the unique limit or fixed point behavior of these models.

Definition B.6 (Irreducibility). A Markov model $\langle \pi, \tau \rangle$ with transition matrix T is irreducible if, for each pair of states $i, j \in \pi$, there is a strictly positive probability of reaching j from i in a finite number of steps, or formally,

$$\forall i, j \in \pi : \exists n \geq 1 : T^n_{ji} > 0. \tag{B.1}$$

The set of consecutive steps that have to be taken to go from one state to another is called a path. Each of these paths (commonly, more than one path allows to step from one state to another) has a path length, i.e., the number of steps needed to finish the path.

Definition B.7 (Set of path lengths). For a Markov model $\langle \pi, \tau \rangle$ with transition matrix T, let the set of path lengths D_i be the set of integers denoting the lengths of possible paths from state i to i, with

$$D_i = \{n \mid T^n_{ii} > 0\}. \tag{B.2}$$

Definition B.8 (Aperiodicity). An irreducible Markov model $\langle \pi, \tau \rangle$ is aperiodic if $\gcd(D_i) = 1$ for all $i \in \pi$.

Theorem B.2. *An irreducible Markov model $\langle \pi, \tau \rangle$ with transition probability matrix T and $\exists i \in \pi : T_{ii} > 0$, is aperiodic.*

Proof. We show that, for an arbitrary $j \in \pi$, $\gcd(D_j) = 1$. Since $\langle \pi, \tau \rangle$ is irreducible, a path exists from state j to i, and back from i to j. The length l of this complete path from j to j via i is in D_j. Since $T_{ii} > 0$, an extra step can be taken to go to i from i, as we arrived in i starting our path in j. Hence, $l + 1$ is also in D_j. Since $\gcd(l, l + 1) = 1$ holds for any strictly positive $l \in \mathbb{N}$, $\gcd(D_j)$ is 1. This holds for any j. Hence, the Markov model is aperiodic. □

B.1.3 The Perron-Frobenius Theorem and its Applications for DTMMs

In DTMMs, a probability distribution $\mathbf{x}_g$ at time g over the state space π is multiplied with the $n \times n$ transition matrix T to find the distribution $\mathbf{x}_{g+1} = T \cdot \mathbf{x}_g$ at the next time step $g + 1$. Given an initial distribution $\mathbf{x}_0$, we can compute the distribution $\mathbf{x}_g$ at time g with $\mathbf{x}_g = T^g \cdot \mathbf{x}_0$. Since T is a stochastic matrix denoting the transition probabilities, every T^g is stochastic. Hence, any $\mathbf{x}_g$ is stochastic because of Theorem B.1. As we become interested in the long term behavior of T, or the convergence of $\mathbf{x}_g$ and existence of $\lim_{g \to \infty} \mathbf{x}_g$, we adopt the following important theorem in Markov chain analysis, which is a specific case of the Perron-Frobenius theorem (Perron 1907; Frobenius 1912).

Theorem B.3 (Perron-Frobenius). *Let T be a stochastic, irreducible and aperiodic $n \times n$ matrix. Then*

1. *T has an eigenvalue $\lambda_1 = 1$ with corresponding stochastic eigenvector,*
2. *for all other eigenvalues $\lambda_g \neq \lambda_1$ of T, $0 \leq |\lambda_g| < \lambda_1 = 1$ holds, and*
3. *eigenvalue λ_1 has multiplicity 1.*

Theorem 1.9.6 in Bapat & Raghavan (1997) proves this theorem. Moreover, Bapat and Raghavan's theorem shows the following:

Theorem B.4. *Let T be a stochastic, irreducible and aperiodic $n \times n$ matrix. Then $\lim_{g \to \infty} T^g = Q$ exists. Matrix Q has identical columns, each column of Q is given by the unique stochastic vector $\mathbf{u}$ that satisfies $T\mathbf{u} = \mathbf{u}$, i.e. the unique stochastic eigenvector of T with corresponding eigenvalue 1.*

The limit distribution of the Markov chain according to the irreducible and aperiodic transition probability matrix T is given by

$$(\mathbf{x}_\infty)_i = \left(\lim_{g \to \infty} T^g \mathbf{x}_0 \right)_i = (Q\mathbf{x}_0)_i = \sum_{j=1}^{n} Q_{ij} (\mathbf{x}_0)_j = u_i \tag{B.3}$$

since $\sum_{j=1} (\mathbf{x}_0)_j = 1$ and all Q_{ij} correspond to elements u_i of eigenvector $\mathbf{u}$ with corresponding eigenvalue 1 of T. Consequently, the limit distribution of an irreducible and aperiodic transition probability matrix is equal to the unique stochastic eigenvector with corresponding eigenvalue 1, independent of the initial distribution $\mathbf{x}_0$,

$$\lim_{g \to \infty} T^g \mathbf{x}_0 = \mathbf{u}. \tag{B.4}$$

Throughout this thesis, we have adopted this framework to find the limit or fixed point behavior of finite Markov models; If the stochastic transition probability matrix is irreducible and aperiodic, then the Markov model's limit behavior can be discussed in terms of its unique stochastic eigenvector with corresponding eigenvalue 1.

B.2 Combinations of Markov Models

Consider the state space π which is the Cartesian product of n sub state spaces π_k, i.e., $\pi = \pi_1 \times \cdots \times \pi_n$. Let $\pi_{\neg k}$ denote the Cartesian product of sub state spaces, leaving out π_k, i.e. $\pi_{\neg k} = \pi_1 \times \cdots \times \pi_{k-1} \times \pi_{k+1} \times \cdots \times \pi_n$. A state $\mathbf{s} \in \pi$ is a vector $\mathbf{s} = (s_1, \ldots, s_n)^\top$ with each $s_k \in \pi_k$. Let $\mathbf{s}_{\neg k} = (s_1, \ldots, s_{k-1}, s_{k+1}, \ldots, s_n)^\top$, i.e. the vector denoting the elements of $\mathbf{s}_k$ in $\pi_{\neg k}$.

Consider n seperate Markov models $\langle \pi_k \times \pi_{\neg k}, \tau_k \rangle$. Transition function τ_k maps elements from and into π_k and leaves the other states in $\pi_{\neg k}$ untouched. This function τ_k is dependent on the state $\mathbf{s}_{\neg k} \in \pi_{\neg k}$ of the other Markov models. Let Markov model $\langle \pi, \tau \rangle$ be the combination of n seperate, interdependent sub Markov models $\langle \pi_k \times \pi_{\neg k}, \tau_k \rangle$ with transition rule

$$\mathbb{P}\left[\tau(\mathbf{s}) = \mathbf{s}'\right] = \prod_{k=1}^{n} \mathbb{P}\left[\tau_k(s_k) = s'_k \mid \mathbf{s}_{\neg k}\right]. \tag{B.5}$$

Under the assumption that any of the sub Markov models is irreducible and aperiodic if all other Markov models remain fixed in their state, we now want to know whether the combined Markov model inherits these properties of irreducibility and aperiodicity.

B.2.1 Combination of Independent Markov models

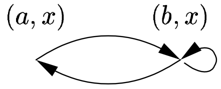
Sub models $\langle \pi_k \times \pi_{\neg k}, \tau_k \rangle$ of $\langle \pi, \tau \rangle$ are independent if $\mathbb{P}\left[\tau_k(s_k) = s'_k \mid \mathbf{s}_{\neg k}\right] = \mathbb{P}\left[\tau_k(s_k) = s'_k\right]$ holds for all $s_k, s'_k \in \pi_k$ and all k.

If all sub models are ergodic, then it is trivial that the combination of these models is also ergodic. Independent of each other, the limit behavior of each model exists and is uniquely defined by the stochastic eigenvector $\mathbf{x}^*_k$ of their respective transition probability matrices. The limit distribution $\mathbf{x}^*$ of the combined Markov model has elements $x^*_{\mathbf{s}} = \prod_{k=1}^{n} (\mathbf{x}^*_k)_{s_k}$.

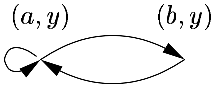
B.2.2 Combination of Dependent Markov models

When considering the co-evolution of two populations in Chapter 7, the transition probabilities of either population is dependent on the state of the other population. Generally, the combination of such dependent ergodic Markov models is not always ergodic, which is shown with the following example.

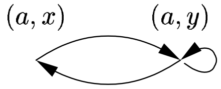
Example B.1. Consider the following 2 dependent systems. The first system has state space $\{a, b\}$ and the second has state space $\{x, y\}$. Consider the digraphs



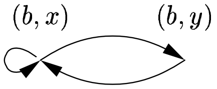
and



representing the state transitions of the first model, if the second system is in state x and y respectively. Vertexes represent the states of the system. Directed edges denote strictly positive transition probabilities. If no directed edge is present, the transition probability is 0. Similarly, consider

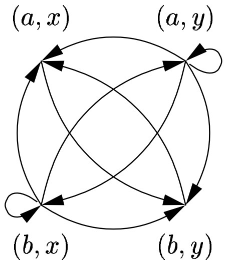


and



to be the state transitions for the second system, if the first system is in state a and b respectively. All these separate models are irreducible and aperiodic.

The combination of these models, represented by the transition graph



,

is however not irreducible. Once the system resides in either state (a, x) or (b, y), it will not be able to reach states (a, y) and (b, x) again, making the model irreducible.

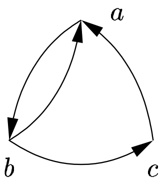
□

As a consequence of this example, it is not generally true that the combination of dependent Markov models with irreducible and aperiodic transition matrices results in an irreducible and aperiodic Markov model.

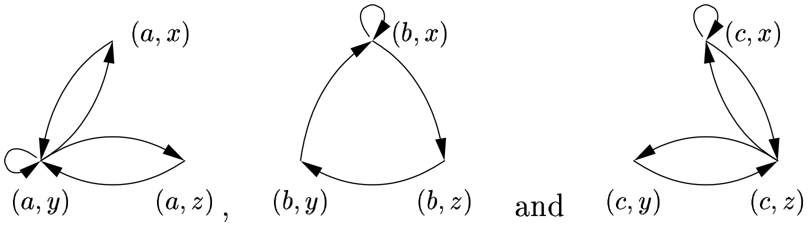
B.2.3 Combination of Semi Dependent Markov models

In Chapter 5 we study populations in an explicitly defined dynamic fitness environment. In the models studied in Section 5.3, the population is dependent on the state of the environment, which is implemented as a Markov chain on its own. The model that represents the environment is not dependent on the state of the population. Similar to the previous section, we are interested in knowing whether such semi dependent combinations of Markov chains – where at least one of the sub models is independent of the other and at least one is dependent on the other – is ergodic, if the constituent sub models are ergodic. However, the following example shows that this is not generally true.

Example B.2. Consider the following 2 semi dependent models, where the second model depends on the state of the first system, and the first model is independent of the second. The first model has state space $\{a, b, c\}$ and the second has state space $\{x, y, z\}$. Let

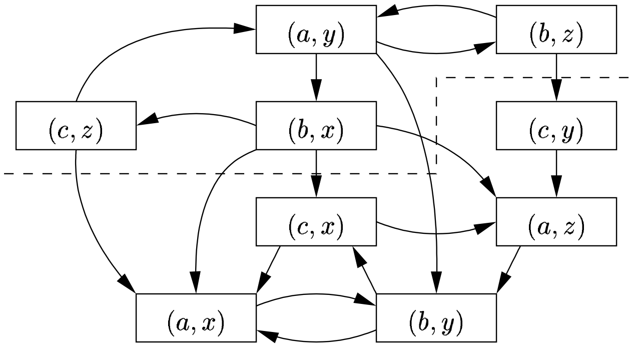


be the digraph that represents transitions in the first model. Let digraphs



describe the transitions of the second model as the first model is in states a, b and c respectively. All of the above independent models are individually irreducible and aperiodic.

The combination of these models is represented by the following digraph.



Note that none of the states above the dashed line can be reached if the system is started in one of the states below the dashed line. Therefore, the resulting model is not irreducible.

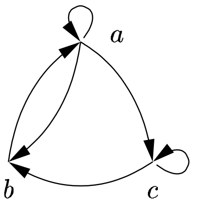
□

As a result from this example, it is not generally true that the semi dependent combination of ergodic Markov models results in a combined Markov model that is ergodic too.

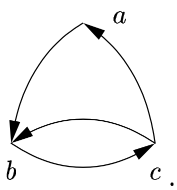
B.3 Alternation of Markov models

Besides constructing combinations of Markov models, we also combine Markov models – defined over the same state space – by multiplying their transition matrices, as in Section 5.2. This allows the separate sub models to govern the dynamics of the system alternately. With an example, we show that the multiplication of ergodic transition probability matrices does not generally result in a transition probability matrix that is ergodic.

Example B.3. Consider two ergodic Markov models over a state space $\{a, b, c\}$, whose transitions are depicted by the digraphs



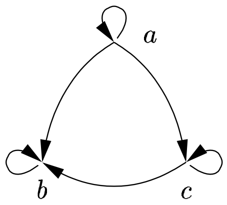
and



.

Both of these models are irreducible and aperiodic.

We allow the combined Markov model to take one step in either model alternately. The digraph that represents the combined Markov model that takes one step in the first, and then a step in the second model can be represented by the digraph



.

This combined model is not irreducible since there is no transition leading away from state b.

□

As a result, the alternation of ergodic Markov models is not necessarily ergodic.

Bibliography

Alkemade, F., J. La Poutré, & H. Amman (2005). On Social Learning and Robust Evolutionary Algorithm Design in Economic Games. In *2005 Congress on Evolutionary Computation (CEC 2005), to appear.*

Altenberg, L. (1995). The Schema Theorem and Price's Theorem. In L. D. Whitley & M. D. Vose (Eds.), *Foundations of Genetic Algorithms 3*, pp. 23–49. Morgan Kaufmann.

Arnold, D. V. & H.-G. Beyer (2002). Random Dynamics Optimum Tracking with Evolution Strategies. In J. Merelo, P. Adamidis, H.-G. Beyer, J. Fernandez-Villacanas, & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature*, pp. 3–12. Springer-Verlag.

Banzhaf, W. (1988). Optimization by Diploid Search Strategies. In *Neural and Synergetic Computers, Proceedings of the International Symposium on Synergetics*, pp. 155–167. Springer-Verlag.

Bapat, R. B. & T. E. S. Raghavan (1997). *Nonnegative Matrices and Applications.* Cambridge University Press.

Barnett, L. (1997). Tangled Webs: Evolutionary Dynamics on Fitness Landscapes with Neutrality. Master's thesis, University of Sussex.

Barrow, B. (1997, January). A Lesson in Megabytes. *IEEE Standards Bearer*.

Bidwell, G. L. (1996). Applying Diploidy and Dominance to Artificial Genetic Search. Master's thesis, University of Montana.

Bierman, H. S. & L. Fernandez (1997). *Game Theory with Economic Applications* (second ed.). Addison Wesley.

Bošnački, D., H. M. M. ten Eikelder, & P. A. J. Hilbers (2003). Genetic Code as a Gray Code Revisited. In *The 2003 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences.*

Branke, J. (2001). *Evolutionary Optimization in Dynamic Environments.* Kluwer.

Branke, J. (2003). Evolutionary Approaches to Dynamic Optimization Problems, Introduction and Recent Trends. In *GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems.*

Branke, J. & W. Wang (2003). Theoretical Analysis of Simple Evolution Strategies in Quickly Changing Environments. In *Proceedings of the 2003 Genetic and Evolutionary Computation Conference.*

Chattoe, E. (1998). Just How (Un)Realistic are Evolutionary Algorithms as Representations of Social Processes? *Journal of Artificial Societies and Social Simulation 1*.

Collingwood, E., D. Corne, & P. Ross (1996). Useful Diversity via Multiploidy. In *IEEE International Conference on Evolutionary Computing*, pp. 810–813.

Cover, T. M. & J. A. Thomas (1991). *Elements of Information Theory*. New York: Wiley.

Cramer, M. L. (1985). A Representation for the Adaptive Generation of Simple Sequential Programs. In J. J. Grefenstette (Ed.), *Proceedings of an International Conference on Genetic Algorithms and their Applications*, pp. 183–187.

Crombach, A. (2002). Engineering a Diploid and Co-Evolutionary Genetic Algorithm. Technical report, Faculteit Biomedische Technologie, Technische Universiteit Eindhoven.

Crow, J. F. & M. Kimura (1970). *An Introduction to Population Genetics Theory*. New York: Harper and Row.

Darwin, C. (1859). *Origin of Species*. Oxford University Press.

De Jong, K. A., W. M. Spears, & D. F. Gordon (1995). Using Markov Chains to Analyze GAFOs. In L. D. Whitley & M. D. Vose (Eds.), *Foundations of Genetic Algorithms III*, pp. 115–137. Morgan Kaufmann.

Deb, K. e. a. (2004). *Proceedings of the Genetic and Evolutionary Computation Conference*. Springer.

Del Corso, G. M. (1996). Estimating an Eigenvector by the Power Method with a Random Start. *SIAM J. on Matrix Anal. Appl. 18*, 913–937.

Del Corso, G. M. (1997). *Randomized Methods for Eigenvalue and Eigenvector Approximation*. Ph. D. thesis, University of Milano.

Dixit, A. K. & B. J. Nalebuff (1993). *Thinking Strategically: The Competitive Edge in Business, Politics, and Everyday Life*. W. W. Norton and Company.

Droste, S. (2002). Analysis of the (1+1) EA for a Dynamically Changing Onemax Variant. In *Congress on Evolutionary Computation*, pp. 55–60.

Eiben, A. E., E. H. L. Aarts, & K. M. Van Hee (1992). Global Convergence of Genetic Algorithms: An Infinite Markov Chain Analysis. In *Proceedings of the First International Conference on Parallel Problem Solving from Nature*, pp. 4–12. Springer.

Eigen, M., J. McCaskill, & P. Schuster (1989). The Molecular Quasispecies. *Advances in Chemical Physics 75*, 149–263.

Ewens, W. J. (2004). *Mathematical Population Genetics* (second ed.). Springer-Verlag.

Feller, W. (1968). *An Introduction to Probability Theory and its Applications* (third ed.), Volume 1. Wiley.

Ficici, S. G. (2004). *Solution Concepts in Coevolutionary Algorithms.* Ph. D. thesis, Brandeis University.

Ficici, S. G., O. Melnik, & J. B. Pollack (2000). A Game-Theoretic Investigation of Selection Methods Used in Evolutionary Algorithms. In *Proceedings of the 2000 Congress on Evolutionary Computation.*

Ficici, S. G. & J. B. Pollack (2000a). Effects of Finite Populations on Evolutionary Stable Strategies. In *Proceedings of the 2000 Genetic and Evolutionary Computation Conference.*

Ficici, S. G. & J. B. Pollack (2000b). A Game-Theoretic Approach to the Simple Coevolutionary Algorithm. In *Parallel Problem Solving from Nature VI.*

Fischer, R. A. (1922). On the Dominance Ratio. *Proceedings of the Royal Society Edinb. 42*, 321–324.

Fischer, R. A. (1930). *The Genetical Theory of Natural Selection.* Clarendon Press, Oxford.

Floreano, D., F. Mondada, & S. Nolfi (1999). Co-evolution and Ontogenetic Change in Competing Robots. In *Robotics and Autonomous Systems.*

Fogel, D. B. & G. B. Fogel (1995). Evolutionary Stable Strategies are not always Stable under Evolutionary Dynamics. In *Evolutionary Programming IV*, pp. 565–577.

Fogel, D. B., G. B. Fogel, & P. C. Andrews (1997). On the Instability of Evolutionary Stable Strategies. *BioSystems 44*, 135–152.

Fogel, G. B., P. C. Andrews, & D. B. Fogel (1998). On the Instability of Evolutionary Stable Strategies in Small Populations. *Ecological Modelling 109*, 283–294.

Foster, D. & P. Young (1990). Stochastic Evolutionary Game Dynamics. *Journal of Theoretical Population Biology 38*, 219–232.

Frobenius, F. G. (1912). Über Matrizen aus nicht negativen Elementen. *Sitzungsberichte der Königlich Preusischen Akademie de Wissenschaften zu Berlin 4*, 456–477.

Fudenberg, D. & D. K. Levine (1998). *The Theory of Learning in Games (Economic Learning and Social Evolution).* MIT Press.

Fudenberg, D. & J. Tirole (1991). *Game Theory.* MIT Press.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional.

Goldberg, D. E. & R. E. Smith (1987). Nonstationary Function Optimization using Genetic Algorithms with Dominance and Diploidy. In *Second International Conference on Genetic Algorithms*, pp. 59–68.

Greene, B. (1999). Crossover and Diploid Dominance with Deceptive Fitness. In *Congress on Evolutionary Computation*, pp. 1369–1376.

Hadad, B. S. & C. F. Eick (1997). Supporting Polyploidy in Genetic Algorithms Using Dominance Vectors. In *Evolutionary Programming VI*, pp. 223–234.

Haldane, J. B. S. (1927). A Mathematical Theory of Natural and Artificial Selection. V. Selection and Mutation. *Proceedings of the Cambridge Philosophical Society 23*, 838–844.

Hardy, G. H. (1908). Mendelian Proportions in a Mixed Population. *Science 28*, 49–50.

Harsanyi, J. C. & R. Selten (1988). *A General Theory of Equilibrium Selection in Games*. MIT Press, Camnbridge, Massachusetts.

Hartl, D. L. (2000). *A Primer in Population Genetics* (third ed.). Sinauer Associates.

Hartl, D. L. & A. G. Clark (1997). *Principles of Population Genetics* (third ed.). Sinauer Associates.

Hillis, D. (1992). Co-evolving parasites improve simulated evolution as an optimization procedure. In *Artificial Life II*. Addison Wesley.

Hofbauer, J. & K. Sigmund (1998). *Evolutionary Games and Population Dynamics*. Cambridge University Press.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Horn, J. (1997). *The Nature of Niching: Genetic Algorithms and the Evolution of Optimal, Cooperative Strategies*. Ph. D. thesis, university of Illinois at Urbana-Campaign.

IEC (2000, November). *60027-2, Letter Symbols to be Used in Electrical Technology - Part 2: Telecommunications and Electronics* (second ed.). International Electrotechnical Commission.

Kandori, M., G. J. Mailath, & R. Rob (1993). Learning, Mutation, and Long Run Equilibria in Games. *Econometrica, Econometric Society 61*, 29–56.

Kimura, M. (1955). Solution of a Process of Random Genetic Drift with a Continuous Model. *National Academy of Science U.S. 41*, 144–150.

Kimura, M. (1957). Some Problems of Stochastic Processes in Genetics. *Annals of Mathematical Statistics 28*, 882–901.

Kimura, M. (1962). On the Probability of Fixation of Mutant Genes in a Population. *Genetics 47*, 713–719.

Kimura, M. (1964). Diffusion Models in Population Genetics. *Journal of Applied Probability 1*, 177–232.

Kimura, M. (1968). Evolutionary Rate at the Molecular Level. *Nature 217*, 624–626.

Kimura, M. (1986). *The Neutral Theory of Molecular Evolution* (reprint ed.). Cambridge University Press.

Kimura, M. (1994). *Population Genetics, Molecular Evolution, and the Neutral Theory: Selected Papers*. University of Chicago Press.

Kimura, M. & T. Otha (1969). The Average Number of Generations Until Fixation of a Mutant Gene in a Finite Population. *Genetics 61*, 763–771.

Kostlan, E. (1988). Complexity Theory of Numerical Linear Algebra. *J. Computational and Applied Mathematics 22*, 219–230.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press.

Kuczyńsky, J. & H. Woźniakowski (1992). Estimating the Largest Eigenvalue by the Power and Lanczos Algorithms with a Random Start. *SIAM J. Matrix Anal. Appl. 13*, 1094–1122.

Kullback, S. (1959). *Information Theory and Statistics*. New-York: Wiley.

Kullback, S. & R. A. Leibler (1951). On Information and Sufficiency. *Ann. Math. Stat. 22*, 79–86.

Langdon, W. B. & R. Poli (2002). *Foundations of Genetic Programming*. Springer-Verlag.

Lewis, J., E. Hart, & G. Ritchie (1998). A Comparison of Dominance Mechanisms and Simple Mutation on Non-stationary Problems. In *Parallel Problem Solving from Nature V*, pp. 139–148.

Liekens, A. M. L., H. M. M. ten Eikelder, & P. A. J. Hilbers (2003a). Finite Population Model Analysis of Co-Evolution with Matching Pennies. In *Fifth International Workshop on Frontiers in Evolutionary Algroithms (FEA 2003)*.

Liekens, A. M. L., H. M. M. ten Eikelder, & P. A. J. Hilbers (2003b). Finite Population Models of Co-Evolution and their Application to Haploidy versus Diploidy. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003)*.

Liekens, A. M. L., H. M. M. ten Eikelder, & P. A. J. Hilbers (2003c). Finite Population Models of Dynamic Optimization with Alternating Fitness

Functions. In *GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems.*

Liekens, A. M. L., H. M. M. ten Eikelder, & P. A. J. Hilbers (2003d). Finite Population Models of Dynamic Optimization with Stochastically Alternating Fitness Functions. In *2003 Congress on Evolutionary Computation (CEC 2003).*

Liekens, A. M. L., H. M. M. ten Eikelder, & P. A. J. Hilbers (2003e). Modeling and Simulating Diploid Simple Genetic Algorithms. In K. De Jong, R. Poli, & J. Rowe (Eds.), *Foundations of Genetic Algorithms VII.*

Liekens, A. M. L., H. M. M. ten Eikelder, & P. A. J. Hilbers (2004). Predicting Genetic Drift in 2×2 Games. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004).*

Lubberts, A. & R. Miikkulainen (2001). Co-Evolving a Go-Playing Neural Network. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, pp. 14–19.

Mathews, J. & R. L. Walker (1964). *Mathematical Methods of Physics.* Addison-Wesley.

Maynard Smith, J. (1982). *Evolution and the Theory of Games.* Cambridge University Press.

Maynard Smith, J. & G. R. Price (1973). The Logic of Animal Conflict. *Nature 246*, 15–18.

Mendel, G. (1865). Versuche über Pflanzen-Hybriden. *Verhandlungen des Naturforshenden Vereins in Brünn 4*, 3–47.

Miękisz, J. (2004a). Statistical Mechanics of Spatial Evolutionary Games. *Journal of Physics A 37*, 9891–9906.

Miękisz, J. (2004b). Stochastic Stability in Spatial Games. *Journal of Statistical Physics 117*, 99–110.

Miękisz, J. (2004c). Stochastic Stability in Spatial Three-Player Games. *Physica A 343*, 175–184.

Miękisz, J. (2005a). Equilibrium Selection in Evolutionary Games with Random Matching of Players. *Journal of Theoretical Biology 232*, 47–53.

Miękisz, J. (2005b). Long-Run Behavior of Games with Many Players (to appear). *Markov Processes and Related Fields.*

Mitchell, M. (1996). *An Introduction to Genetic Algorithms.* MIT Press.

Moey, C. C. J. & J. E. Rowe (2004a). Population Aggregation Based on Fitness. *Natural Computing 3*(1), 5–19.

Moey, C. C. J. & J. E. Rowe (2004b). A Reduced Markov Model of GAs without the Exact Transition Matrix. In H.-P. Schwefel (Ed.), *Proc. of the 8th Intl. Conference on Parallel Problem Solving from Nature (PPSN VIII)*. Springer-Verlag.

Moran, P. A. (1962). *The Statistical Processes of Evolutionary Theory.* Clarendon, Oxford.

Naudts, B. (1998). *Measuring GA-hardness.* Ph. D. thesis, University of Antwerp.

Naudts, B. & A. Verschoren (1999). Epistasis and Deceptivity. *Bulletin of the Belgian Mathematical Society - Simon Stevin 6*, 147–154.

Neill, D. B. (2004). Evolutionary Stability for Large Populations. *Journal of Theoretical Biology 227*, 397–401.

Ng, K. P. & K. C. Wong (1995). A New Diploid Scheme and Dominance Change Mechanism for non-Stationary Function Optimization. In *6th Int. Conf. on Genetic Algorithms*, pp. 159–166.

Ng, P. C. & S. Henikoff (2002). Accounting for Human Polymorphisms Predicted to Affect Protein Function. *Genetics 12*, 436–446.

Nix, A. E. & M. D. Vose (1992). Modelling Genetic Algorithms with Markov Chains. *Annals of Mathematics and Artificial Intelligence*, 79–88.

Nowak, M. A., A. Sasaki, C. Taylor, & D. Fudenberg (2004). Emergence of Cooperation and Evolutionary Stability in Finite Populations. *Nature 428*, 646–650.

Nowak, M. A. & K. Sigmund (2004). Evolutionary Dynamics of Biological Games. *Science 303*, 793–799.

Oberhumer, M. F. X. J. (2002). *LZO Real-Time Data Compression Library, http://www.oberhumer.com/opensource/lzo/.*

Papoulis, A. & S. Unnikrishna (2002). *Probability, Random Variables, and Stochastic Processes* (fourth ed.). New York: McGraw-Hill.

Parlett, B. N. (1980). *The Symmetric Eigenvalue Problem.* Prentice-Hall.

Perron, O. (1907). Über Matrizen. *Mathematische Annalen 64*, 248–263.

Price, G. (1970). Selection and covariance. *Nature 227*, 520–521.

Price, G. (1972). Fisher's "Fundamental Theorem" Made Clear. *Annals of Human Genetics 36*, 129–140.

Prügel-Bennett, A. (2003). Modelling Finite Populations. In K. de Jong, R. Poli, & J. E. Rowe (Eds.), *Foundations of Genetic Algorithms VII*, pp. 99–114.

Qian, H. (2001). Relative Entropy: Free Energy Associated with Equilibrium Fluctuations and Nonequilibrium Deviations. *Physical Review E 63*, 042103/1–042103/4.

Radcliffe, N. J. (1992). Non-Linear Genetic Representations. In R. Männer & B. Manderick (Eds.), *Parallel problem solving from nature 2*, pp. 259–268. North-Holland.

Riley, J. G. (1997). Evolutionary Equilibrium Strategies. *Journal of Theoretical Biology*, 109–123.

Robson, A. J. & F. Vega-Redondo (1996). Efficient Equilibrium Selection in Evolutionary Games with Random Matching. *Journal of Economic Theory 70*, 65–92.

Ronnewinkel, C., C. O. Wilke, & T. Martinetz (2001). Genetic Algorithms in Time-Dependent Environments. In L. Kallel, B. Naudts, & A. Rogers (Eds.), *Theoretical Aspects of Evolutionary Computation*, pp. 261–285.

Rosin, C. D. (1997). *Coevolutionary Search Among Adversaries.* Ph. D. thesis, University of California at San Diego.

Rowe, J. E. (2001). Cyclic Attractors and Quasispecies Adaptability. In L. Kallel, B. Naudts, & A. Rogers (Eds.), *Theoretical Aspects of Evolutionary Computation*, pp. 261–285.

Rowe, J. E., M. D. Vose, & A. D. Wright (2005). State Aggregation and Population Dynamics in Linear Systems (to appear). *Artificial Life.*

Rudolf, G. (1998). Finite Markov Chain Results in Evolutionary Computation: A Tour d'Horizon. *Fundamenta Informaticae 35*, 67–89.

Ryan, C. & J. J. Collins (1998). Polygenic Inheritance - A Haploid Scheme that can Outperform Diploidy. In *Parallel Problem Solving from Nature V*, pp. 178–187.

Saad, Y. (1992). *Numerical Methods for Large Eigenvalue Problems.* Manchester University Press.

Schaffer, M. E. (1988). Evolutionary Stable Strategies for a Finite Population and Variable Contest Size. *Journal of Theoretical Biology 132*, 469–478.

Schmitt, L. M. (2001). Theory of Genetic Algorithms. *Theoretical Computer Science 259*, 1–61.

Shub, M. (1986). The Geometry and and Topology of Dynamical Systems and Algorithms for Numerical Problems. In L. Shantao (Ed.), *Proceedings of 1983 Beijing Symposium on Differential Geometry and Differential Equations.*

Smith, R. E. & D. E. Goldberg (1992). Diploidy and Dominance in Artificial Genetic Search. *Complex Systems 6*, 251–285.

Spears, W. M. (1996). Analyzing GAs Using Markov Models with Semantically Ordered and Lumped States. In R. K. Belew & M. Vose (Eds.), *Foundations of Genetic Algorithms IV*.

Spears, W. M. (1998a). A Compression Algorithm for Probability Transition Matrices. *SIAM Matrix Analysis and Applications 20*(1), 60–77.

Spears, W. M. (1998b). *The Role of Mutation and Recombination in Evolutionary Algorithms*. Ph. D. thesis, George Mason University.

Spears, W. M. (1999). Aggregating Models of Evolutionary Computation. In P. J. Angeline & et al (Eds.), *Conference on Evolutionary Computation*.

Suys, D. (1999). *A Mathematical Approach to Epistasis*. Ph. D. thesis, University of Antwerp.

Taylor, C., D. Fudenberg, A. Sasaki, & M. A. Nowak (2004). Evolutionary Game Dynamics in Finite Populations. *Bulletin of Mathematical Biology 66*, 1621–1644.

Taylor, P. D. & L. B. Jonker (1978). Evolutionary Stable Strategies and Game Dynamics. *Mathematical Bioscience 40*, 145–156.

Tesfatsion, L. (2001). Special Issue on Agent-Based Computational Economics. *Journal of Economic Dynamics and Control 25*, 281–293.

van Heeswijk, M. (2004). Implementation of a Parallel Framework for Markov Chain Eigenvector Computation (internship report). Technical Report BMI 04/15, Faculteit Biomedische Technologie, Technische Universiteit Eindhoven.

van Kemenade, C. H. M., J. N. Kok, J. A. La Poutré, & D. Thierens (1998). Transmission Function Models of Finite Population Genetic Algorithms. Technical Report SEN-R9893, CWI.

van Nimwegen, E. & J. P. Crutchfield (1998). Optimizing Epochal Evolutionary Search: Population-Size Independent theory. Technical Report 98-06-046, Santa Fe Institute.

van Nimwegen, E., J. P. Crutchfield, & M. Mitchell (1997). Finite Populations Induce Metastability in Evolutionary Search. *Physics Letters A 229*, 144–150.

van Nimwegen, E., J. P. Crutchfield, & M. Mitchell (1999). Statistical Dynamics of the Royal Road Genetic Algorithm. *Theoretical Computer Science, Special Issue on Evolutionary Computation 229*, 41–102.

Vickery, W. L. (1987). How to Cheat against a Simple Mixed Strategy ESS. *Journalof Theoretical Biology 127*, 133–139.

Vitányi, P. (2000). A Discipline of Evolutionary Programming. *Theoretical Computer Science 241*, 3–23.

Vose, M. D. (1999a). Random Heuristic Search. *Theoretical Computer Science 229*, 103–142.

Vose, M. D. (1999b). *The Simple Genetic Algorithm.* MIT Press, Cambridge, Massachusetts.

Vose, M. D. & G. E. Liepins (1991). Punctuated Equilibria in Genetic Search. *Complex Systems 5*, 31–44.

Weibull, J. W. (1995). *Evolutionary Game Theory.* MIT Press, Cambridge, Massachusetts.

Weinberg, W. (1908). On the Detection of Heredity in Man (German). *Jahreshefte der Vereins für Väterlandische Naturkunde in Wurttemberg 64*, 368–382.

Whittaker, E. T. & G. N. Watson (1990). *A Course in Modern Analysis* (fourth ed.). Cambridge University Press.

Wiegand, R. P. (2004). *An Analysis of Cooperative Coevolutionary Algorithms.* Ph. D. thesis, George Mason University.

Wilkinson, J. H. (1965). *The Algebraic Eigenvalue Problem.* Oxford university Press.

Wright, A. H. & G. L. Bidwell (1996). Another Look at Diploidy and Genetic Algorithms.

Wright, S. (1931). Evolution in Mendelian Populations. *Genetics 16*, 97–159.

Yoshida, K. & N. Adachi (1994). A Diploid Genetic Algorithm for Preserving Population Diversity. *Parallel Problem Solving from Nature III*, 36–45.

Nederlandse Samenvatting

Een belangrijke parameter van het evolutionaire gedrag van een populatie is zijn grootte. In grote populaties kunnen een groot aantal gencombinaties en genvariaties uitgeprobeerd worden. Zelfs als er nadelige mutaties optreden dan sterven deze gemakkelijk uit door de grote hoeveelheid individuen met een selectief voordeel in de populatie. De nadelige effecten van willekeur in grote populaties zijn daardoor meestal verwaarloosbaar.

Stochastische effecten zijn echter van belang in kleine populaties. De kans dat een *ongelukkige* verzameling allelen worden overgeërfd door een populatie is groter als de groep nakomelingen kleiner is. Het is een bekend feit van de statistiek dat kleine samples niet representatief zijn. Gelijkaardig zijn de proporties van allelen in een kleine populatie nakomelingen niet representatief voor de samenstelling van de ouderpopulatie. In een kleine populatie kunnen zo'n onregelmatige proporties zich generatie na generatie opstapelen, zodat allelen verloren kunnen gaan. Dit verschijnsel heet genetische drift, en leidt tot fixatie van de populatie op een beperkt aantal allelen. Door minder diversiteit in een kleine populatie gaat het vermogen om zich aan te passen aan de omgeving achteruit. Meer nog, de omgeving van de populatie kan veranderen over de generaties heen. Door het verlies van diversiteit kan het voor een kleine populatie zeer moeilijk worden zich tijdig aan te passen aan zo'n veranderende omgeving.

In dit proefschrift beschrijven we wiskundige modellen om de evolutie van kleine populaties te bestuderen in een waaier van dynamische omgevingen. De gebruikte Markov modellen beschrijven het gedrag van eindige populaties van abstracte individuen, en laten zien hoe zo'n populatie zich gedraagt na een groot aantal generaties. Met de gebruikte modellen kunnen we dan bestuderen welke invloed de verschillende parameters – zoals de populatiegrootte, mutatiesnelheid, ploïdie, selectieve druk, en de snelheid van veranderingen in de omgeving – hebben op het evolutionaire gedrag van de populatie.

We hebben modellen van eindige populaties in drie verschillende dynamische omgevingen bestudeerd. In het eerste model hebben we dynamische omgevingen gebouwd door actief in te grijpen in de omgeving op gegeven tijdstippen. Op deze manier kunnen we de snelheid van omgevingsveranderingen zelf bepalen, en kunnen we de invloed hiervan bestuderen op het evolutionaire gedrag van de populatie. In het tweede model hebben we technieken van evolutionaire speltheorie gebruikt om dynamische omgevingen te bouwen. Elk van de individuen in de populatie codeert een strategie voor een spel. De individuen in de populatie spelen tegen elkaar, en goede spelers hebben een grotere kans hun strategie door te geven aan de volgende generatie. In dit model is de fitness van een individu dus afhankelijk van de aanwezige strategieën in de populatie. Omdat de samenstelling van de populatie evolueert over de generaties heen bepaalt deze zelf de veranderingen in de omgeving. In het derde model hebben we co-evolutie gemodelleerd door meerdere populaties afhankelijk van elkaar te maken. Door de evolutionaire vooruitgang in een populatie van dit model verandert de omgeving van de andere populaties.

In dynamische omgevingen zijn de effecten van willekeurig samplen en genetische drift van groter belang dan in omgevingen die statisch zijn. Tijdens een periode waar de omgeving relatief statisch blijft kan de populatie voordelige allelen ontdekken, en krijgt de populatie de kans zich te fixeren op deze allelen. Maar als de omgeving dramatisch gewijzigd wordt na zo'n periode van stasis zal de kleine populatie niet de benodigde genetische diversiteit vertonen om zich snel aan de vernieuwde omgeving aan te passen. Bij elke omgevingsverandering zal de populatie zich opnieuw uit zijn toestand van fixatie moeten bevrijden, om de voordelige allelen van de nieuwe omgeving te ontdekken. Dit resulterende proces van steeds wederkerende fixatie en de omgevingsdruk om uit deze toestand te ontsnappen vertraagt de evolutionaire vooruitgang van de kleine populatie.

Om zich steeds te kunnen blijven aanpassen aan de dynamische omgeving heeft de kleine populatie een voldoende grote mutatiesnelheid nodig. Mutaties hebben een grotere diversiteit tot gevolg, en bieden dus een mogelijkheid aan om uit de fixatie te ontsnappen. Als de omgeving sneller verandert, of als de populatie kleiner is, is een hogere mutatiesnelheid gewenst.

Er wordt dikwijls de hypothese gesteld dat populaties van diploïde individuen beter voorzien zijn om omgeveningsveranderingen op te vangen in vergelijking met populaties van haploïde individuen. Deze hypothese stoelt op het feit dat een diploïde populatie meer diversiteit kan vertonen, maar ook omdat de systemen van dominantie en recessiviteit de mogelijkheid bieden een impliciet geheugen op te bouwen met oplossingen voor vroegere omgevingen. De meeste observaties in dit proefschrift tonen echter geen significante voordelen van diploïdie aan in de bestudeerde dynamische omgevingen. Er moet wel opgemerkt worden dat het abstractieniveau in onze modellen te beperkt kan zijn om de voordelen van diploïdie in dynamische omgevingen tot uiting te laten komen. Aanvullingen aan het huidig model kunnen mogelijk tot betere inzichten in de gestelde hypothese leiden.

De observatie dat eindige populaties in dynamische omgevingen steeds weer gefixeerd geraken, en uit deze toestand van fixatie moeten ontsnappen, is analoog aan de observatie dat eindige populaties toestanden van metastabiliteit uitlokken in statische omgevingen. In een statische omgeving geraken eindige populaties tijdelijk gefixeerd in genotypes alvorens genotypes met hogere fitness te ontdekken. De voortdurende omgevingsveranderingen lokken in een dynamische omgeving zulke gepunctueerde evenwichten uit. Het is gebleken dat spelen met cyclisch gedrag een goed werkbaar alternatief kunnen bieden voor het bestuderen van deze gepunctueerde evenwichten.

Woord van dank

Dat enkel mijn naam op de voorpagina van dit boekje prijkt, is heel bedrieglijk. Enkel dankzij de intellectuele en vriendschappelijke steun die ik van mijn omgeving heb genoten is dit boekje een feit kunnen worden.

Vooreerst wil ik mijn promotor, Peter Hilbers, bedanken voor de aangename onderzoeksgroep waarbinnen ik de laatste jaren heb mogen werken. Je hebt me de vrijheid gegund om een eigen onderzoek op poten te zetten, en hebt steeds een kritische interesse getoond voor mijn werk. Ik ben heel wat verschuldigd aan Huub ten Eikelder. Belangrijke stukken van dit proefschrift hebben we samen bedacht en uitgewerkt. Je had slechts enkele woorden nodig om mijn vragen te begrijpen, en vol enthousiasme mee op zoek te gaan naar het correcte antwoord. Ik kan me geen betere directe begeleiding inbeelden. Mijn tweede promotor, Emile Aarts, en de leden van de commissie wil ik bedanken voor hun interesse in mijn werk, en de commentaren en opmerkingen die ze plaatsten bij mijn onderzoek. De stage- en afstudeerstudenten die ik heb mogen begeleiden, Anton, Mario, Mark en Laura, wil ik danken voor hun input. Een aantal belangrijke resultaten zijn hier aanwezig door jullie medewerking. Dankzij de collega's, Ronald, Bart, Koen, Nico, Dragan, Peter, Luc, René en Margret, en de studenten in het lab ben ik steeds met veel plezier naar Eindhoven afgezakt. Dank je wel dat jullie mijn uitgesproken Vlaamse vaderlandsliefde hebben willen doorstaan.

I would like to thank all of my international colleagues and friends in the theoretical GA community for their input, comments, and discussions. This thesis was developed with free and open source software. I would like to thank all of the anonymous developers.

Walle, Manu en Dale, graag had ik jullie willen danken voor de vriendschap die we de laatste jaren hebben gedeeld, en het dagelijks contact dat we met elkaar hebben. Dank je wel voor de technische ondersteuning en het wekelijkse afkoelen boven de potten in de kookles.

Ik wil dit boekje opdragen als herinnering aan Mama. Papoe, graag wil ik je danken voor de kracht die je ons de laatste jaren hebt getoond. Je stond steeds klaar voor ons, geloofde in ons, en wist ons op de juiste momenten te stimuleren om ons best te doen. Annemie en Johan, ook jullie steun is Els en mezelf de laatste jaren niet onopgemerkt voorbijgegaan.

Els, jij krijgt de ereplaats. Je hebt jezelf een paar maanden aan de kant willen zetten om me de rust en tijd te geven die ik voor dit boekje nodig had. Ook al omdat je zwanger was van onze Staf wil ik je vanuit het diepste van mijn hart bedanken voor al de inspanningen die je voor me over had. Ik beloof je dat ik nu heel wat meer tijd ga hebben voor ons gezinnetje.

Dank je wel allemaal,

Curriculum Vitae

Anthony Liekens werd op 12 december 1975 geboren in Sint-Niklaas, België. Na het behalen van het TSO diploma bij de Broeders Hiëronymieten te Sint-Niklaas volgde hij een jaar bijzonder wetenschappelijke vorming aan het Sint-Jan Berchmanscollege in Antwerpen. Hij begon zijn universitaire studie met de kandidaturen Informatica aan de Universiteit Antwerpen. De licentie Informatica heeft hij daarna gevolgd aan de Vrije Universiteit Brussel. Tijdens het laatste jaar van de opleiding heeft hij zijn keuzevakken opgevuld met theoretisch biologische vakken van de faculteit Biologie aan de Universiteit Antwerpen. In oktober 2000 begon hij als assistent in opleiding aan de Technische Universiteit Eindhoven, eerst bij de faculteit Wiskunde en Informatica, daarna bij de faculteit Biomedische Technologie. Gedurende zijn promotie bestudeerde hij Markov modellen van populaties die evolueren in dynamische omgevingen. Op dit moment werkt hij als postdoctoraal onderzoeker bij de faculteit Biomedische Technologie aan de evolutie van metabolische celsignaalnetwerken op basis van een subsidie van de Europese Unie.