CHALMERS | UNIVERSITY OF GOTHENBURG

Evolution towards optimality in Urdar – an artificial ecosystem

Master of Science Thesis

VIKTOR JONSSON

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Göteborg, Sweden, 2011

EVOLUTION TOWARDS OPTIMALITY IN URDAR – AN ARTIFICIAL ECOSYSTEM
Viktor Jonsson

Examiner: Torbjörn Lundh

Department of Mathematical Sciences
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Telephone + 46 (0)31-772 1000

Abstract

Cross-feeding is an important aspect of many ecosystems. It has been show to occur among microbes both in natural systems and in bioreactors. Urdar is an artificial life model of an ecosystem where cross-feeding is abundant. Populations in the system are made up from the 256 elementary cellular automaton rules who digest binary strings. The energy needed by the rules to reproduce is taken as the Shannon entropy of the strings. Rules that are able to increase the entropy of the strings they digest are allowed to divide. The question behind the project is: Will a system with selection for energy consumption on the individual be optimized on the population level towards high energy consumption? With basis is Urdar this question is tackled both on the global and local level. The main result is that the final stable population in the system tend to have high efficiency but be separate from the most efficient population composition possible. A proposed way of reducing the system to a set of differential equations is also examined and tested. This might lead to a more generalized method of describing the system that can also be applied to other similar models.

Acknowledgements

First I would like to thank my supervisor Philip Gerlee for his constant support and clear thinking during this project. I would also like to thank my examiner and co-supervisor Torbjörn Lundh for his contagious enthusiasm and many ideas. I would like to thank both for taking the time needed to let me to this project.

Contents

1 Introduction

This section first places the thesis into a biological context. It then reviews a part of the related work made in similar areas. Finally it states the aims of the project.

1.1 Biological background

Cross-feeding or syntrophy is an important feature in many ecosystems. In its basic form cross-feeding is the situation where one species generate waste that another species can digest further. Cross-feeding can occur in one direction or both, with both species utilizing some intermediary metabolite secreted by the other, or even involve multiple species in a more complex food-web or resource cycling chain. The most abundant examples are found in microbial systems. Examples include methanogenic enviroments (Hallam et al., 2004), (Stams, 1994) and the human gut (Belenguer et al., 2006). It has been observed to be a factor in the degradation of xenobiotic compounds (Dejonghe et al., 2003), (Katsuyama et al., 2009). Consortia of bacteria has also been investigated with the aim of biologically separate metals (Rawlings and Johnson, 2007).

Optimality in evolution touches upon many questions. It has often been used to make predictions behavior through models with assumptions such as the energy consumption is minimized or that the biomass production is maximized and to understand adaptation (Maynard Smith, 1978), (G.A. and Maynard Smith, 1990). Optimality models in this sense have been criticized for being adaptionistic but have been useful in many cases (Dupré, 1987). The focus on optimality in this project is not on prediction or characterization of individual traits. Instead the question is if productivity in an organism is selected for the level of the individual can the population as a whole be expected to evolve towards an productivity maximum?

This notion of global optimality touches upon the age old debate of kin selection versus group selection. Kin selection is the concept that an organism that carry traits that benefit close relatives will ensure the survival of those traits are these are likely also carried by the near relatives. The famous quote by evolutionary biologist J.B.S. Haldane in the 1930s state "I would lay down my life for two brothers or eight cousins". This is meant as a break even point where the loss of his own genetic material is in evolutionary terms paid for in saving his two brothers or eight cousins. The theory of kin selection was formalized in the 1960s by W.D. Hamilton (Hamilton, 1964). Group selection states that a trait that favors a population as a whole will spread in the population regardless of it's effect in individual fitness. It was strongly debated however if this could be a plausible mechanism for evolution and recent development in this area (Nowak, 2006) has rekindled this debate.

Gaia theory is another way of viewing ecosystems as a whole first introduced by James Lovelock (Lovelock, 1972). The background of the theory is the apparent self organizing and self regulating properties of the biosphere as a whole. The gaia hypothesis in its radical strong formulation it is teleological and argues that there is a purpose and aim behind evolution to explain the apparent ordering in nature. In the weak formulation it is a more general formulation that natural system exhibit self regulating behaviour. Even this observation has been criticized however (Kircher, 2002). The study of self regulating systems is still an important topic. One of the most famous models of such an system is Daisyworld (Watson and Lovelock, 1983) where a world populated white and black daisies self regulate global temperature through their albedo. This kind of system thinking can also be applied to consider productivity in an ecosystem Philips (2008).

1.2 Previous work

The study of artificial life is centered around using often simplified models of living systems to generate understanding of mechanisms underlying biological phenomena (Dittrich et al., 2001). The concept of cross-feeding has been studied previously using models of varying complexity. The model of greatest importance for this project is of course the Urdar model(Gerlee and Lundh, 2010a) but it will be covered in greater detail in section 3.

In (Bull and Harcombe, 2009) and (Estrela and Gudelj, 2010) cross-feeding between two species is studied in a situation resembling growth in bacterial colonies or a batch culture. The model describes the growth of the two species (X and Y) from low density until a set carrying capacity(K) of that colony. Each species has it's own intrinsic growth rate but are also affected by the other species through a term describing cross-feeding. This is summarized in the two following equations.

$$\dot{X} = X(r_x + b_{yx}\frac{Y}{X + c_x})(1 - \frac{X+Y}{K}), \tag{1}$$

$$\dot{Y} = Y(r_x + b_{xy}\frac{X}{Y + c_y})(1 - \frac{X+Y}{K}).$$

The last parenthesis correspond to the carrying capacity. The second factor describes the growth rate with r_x and r_y being the intrinsic growth rate and b_{xy} and b_{yx} the contribution to growth by cross feeding with the other species.

The argument is that by engaging in cross-feeding one species sacrifices part of its own growth(for example excreting metabolites that have not been fully digested) and through this boosts the growth of the other species. Through this the first article(Bull and Harcombe, 2009) examines the possibilities for cross-feeding to evolve. It also investigates the stability of an evolved population of cross-feeders by looking at the susceptibility to exploiters or cheats. The definition of an exploiter in this case is a species that can reap the benefits from cross-feeding of the other species but that does not pay the cost of giving resources in return. They find that cheaters will introduced to a population of cross-feeders will eventually take over the population and destroy the cooperation. The authors note three key points affecting the evolution of cross-feeding in their model: the need for spatial heterogeneity, the preexistence of cross-feeding in one direction is required for bidirectional cross-feeding to evolve and there must be a sufficiently high benefit to cost ratio for cross-feeding. In (Estrela and Gudelj, 2010) they expand on the previous model but contrary to the first they reach the conclusion that cross-feeding populations are more resistant to invasion by exploiters than previously thought. This result relies on the assumption that both cross-feeders and exploiters share the same carrying capacity.

In (Pfeiffer and Bonhoeffer, 2004) the model captures the speciation of one bacterial strain into two cross-feeding strains during growth in a chemostat. A chemostat is a bioreactor where there is a constant flow of fresh medium into the reactor and cell culture out of the reactor enabling the system to operate at steady state. The basis for cross-feeding lies in a single reaction with a single intermediate where a strain can either completely degrade the resources flowing into the system or excrete the intermediate allowing another strain to sustain itself from the intermediate. In this model the cross-feeding flows in one direction but there is an assumed penalty to growth rate when the intermediate is present in the reactor. By degrading the intermediate the second strain gives the first an indirect benefit thus making the relationship mutualistic. There is also a cost related to having high levels of enzymes. The key results presented is that the evolution of cross-feeding depends on the dilution rate in the system and the magnitude of the benefits of cross-feeding. The growth rate in this system is described by the following equation.

$$W = f(J^{ATP}) - \sum_{(i=1,\ldots,m+2)} (A_i E_i) - \sum_{(i=1,\ldots,m)} (B_i X_i). \tag{2}$$

The growth rate W is the used to determine growth for each species(N_i) in the system by relating it to the dilution rate(D), $\frac{dN_i}{dt} = (W_i - D)N_i$. What is important is that the levels of intermediates that are the basis for cross-feeding are modeled explicitly. As opposed to being an implicit cost or benefit as in the previous model.

A third way of modeling cross feeding is presented in Combach and Hogeweg (2009). The approach taken does not use differential equations but instead directly simulates a system of digital organisms. The focus lies on cycling resources in a closed system meaning that no resources are added during the simulation. Organisms live in a spatial setting and interact with their nearest neighbors on a grid. The setup produces behavior ranging from single individuals being able to recycle their own resources to cross-feeding patterns. The level of cross-feeding is influenced by the spatial limitations. If organisms are fixed they become very adept at handling their local environment and are more prone to self recycling. When allowed to move around on the grid they are less likely to encounter their own waste and cross-feeding patterns occur more often. The evolution of cross-feeding in the model is also highly dependent on the selection pressure. At high selection pressure rapid cycling of resources is favored. Cross-feeding is generally faster then self-recycling in the model and is favored at high selection pressure.

1.3 Problem formulation

Consider a population of organisms that consume some resource. An organism that is better than average at extracting energy from this resource is more likely to survive and reproduce. The resources that sustain the population can be digested in several steps and cross-feeding is an important factor in the system. The question is now: Will selection on the individual level for organisms that are efficient at extracting energy maximize the whole populations ability to extract energy?

Urdar constitutes such a system and will be studied in an effort to answer the question. A big part of this project is also devoted to generating understanding of the Urdar model itself as some of its behavior is still uncharacterized. Even though some design choices in Urdar might make the analysis of results difficult the Urdar model is taken as is and the underlying dynamics will not be changed during this project.

2 Theory

This section provides the theoretical background for a few concepts used later in the report. First with a more general introduction for each topic and later with focus on the key concepts relevant to this project.

2.1 Cellular automata

The term cellular automata refer to a wide range of mathematical and computational models. They are discrete in time and space usually a grid. Each grid cell has one of a finite set of states. The system is governed by rules that determines the state of the system based on past states. This formalism was first introduced by Ulam and Von Neumann (Ulam, 1962) (von Neumann and Burks, 1966).Famous examples of such systems are John Conway's game of life(Gardner, 1970) and von Neumann's universal constructor (von Neumann and Burks, 1966).

The group of CAs relevant to this project are elementary cellular automata. These constitute the simplest cellular automata system that still exhibit not trivial behavior. They have received much attention in recent years for this reason, a revival that was lead by Stephen Wolfram (Wolfram, 1984). ECAs are one dimensional two state automata. The update rules look at a cell and it's two nearest neighbors to determine the state of that cell in the next time step. With three cells determining the next state the number of possible two state neighborhoods is $2^3 = 8$. A rule is uniquely defined by how it maps these 8 neighborhoods to one or zero making for a total of $2^8 = 256$ possible rules. By the standard notation introduced by Wolfram these are numbered by the eight bit string the mappings represent. For example rule zero(00000000) maps all neighborhoods to zero and rule 255 (11111111) maps all neighborhoods to one. As an example of the standard notation and time evolution of an elementary cellular automaton a few time steps of rule is shown in figure 1.

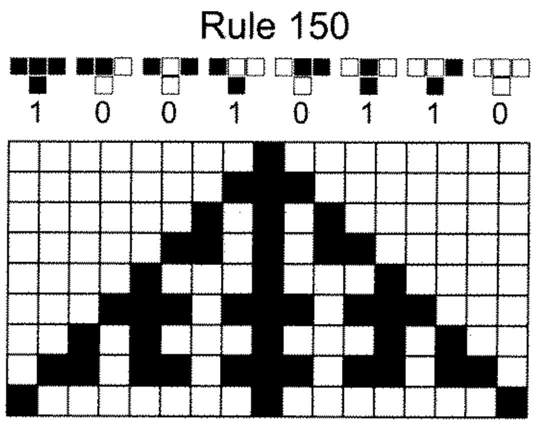


Figure 1: Rule 150 as an example of the basic structure elementary cellular automata. The top part correspond to how neighborhoods map to the bits in the next time step. The bottom part is an example of the time evolution for a few time steps starting from a single one among many zeros.

There are many possible ways to categorize ca-rules. The most important with regards to this project are pairs of inverted rules. These are obtained by flipping all the bits in the output pattern. For example rule zero and 255 are inverted. This is important because under the concept of entropy discussed in the following section inverted strings are equivalent.

2.2 Information theory

Information theory deals with quantifying the information content of systems ranging from binary strings to quantum physics. It was first developed by Claude E. Shannon as a way to characterize the actual data content in signals in an effort to find limits to compression potential (Shannon, 1948). A uniform bit-string of all zeros ...0000... has very low information content. A repeating pattern such as ...010101... has slightly higher information content and ...010011000111... has even more. A completely randomized string has maximum information content and there is no way to describe that random string other than repeating that exact sequence.

In Urdar the organisms consume bit strings and the measure used to quantify their energy content is the Shannon entropy. To define Shannon entropy first consider the concept of block entropy noting that the entropy can be calculated at different length scales or block sizes. For block size one a string need an equal number of ones and zeros to maximize entropy but for block size two it needs and uniform distribution over 00, 01, 10, and 11 to get high entropy. Formally the block entropy S_m for a block of size m is defined as:

$$S_m = \sum_{\sigma_m} p(\sigma_m) log_2 \frac{1}{p(\sigma_m)}. \tag{3}$$

Where σ_m denotes the possible substrings of size m and $p(\sigma_m)$ the probability to find them. The Shannon entropy is then defined as the difference in block entropy, ΔS_m, between blocks of size m and $m+1$ as m goes to infinity:

$$s = \lim_{m \to \infty} \Delta S_m. \tag{4}$$

As also noted in (Gerlee and Lundh, 2010a) the real Shannon entropy isn't used in actual simulations since the strings used are of finite size and for the high computational cost involved. The level at which entropy is calculated in the current implementation is two.

2.3 Theory of many player games

A very successful approach to characterizing situations with interacting individuals has been to formulate these situations in a game theoretic form. Game theory was first developed in the first half of the twentieth century with major contributions by John Nash, John von Neumann and Thomas Schelling (Nash, 1950), (von Neumann, 1928) and (Schelling, 1980). Since then it has then seen applications in a wide array of fields from economics to biology. It has been especially prominent as a tool in evolutionary biology (Maynard Smith, 1982).

The idea behind game theory is to reduce a problem to a set of possible actions, each player picks an action or strategy and receives payoff based on the choice of strategies by the other players. The simplest form of game is a two-player game with two possible actions. This can be formulated into a pay-off matrix as seen in fig something. Each player simultaneously pick an action and receives payoff based as listed by the matrix. The assumption is then that players are rational and strive to maximize their own pay-off. Given this it is possible to argue what actions the player will and ought to take. The most widely used and well know of such characterizations is the Nash equilibrium formulated by John Nash. When the all players play the Nash equilibrium none of the individual players can improve their payoff by choosing another action i.e. the Nash equilibrium represents a stable fixed point of that game. A game may have several Nash equilibria and a Nash equilibrium is not necessarily the set of strategies that maximizes the total payoff as exemplified by the prisoner's dilemma game depicted in figure.

An extension of regular game theory is evolutionary game theory. Instead of players participating in a one shot game there is a population of different strategies that compete in an evolutionary perspective. The strategies who do well increase their fraction and the strategies that do poorly decrease. In evolutionary game theory the interest often lies in finding the evolutionary stable strategy or ESS. The ESS can either be a single strategy or a population consisting of a mix of strategies. The requirement is that no rare mutant is able to invade a population playing the ESS.

2.3.1 Non trivial equilibria in evolutionary multiplayer games

In section Urdar is reformulated in game theoretic terms. This is done in an effort to reduce the simulated dynamics to a more manageable analytical formulation. The objective to characterize the cross-feeding and cooperation between rules still remain. In (Gokhale and Traulsen, 2010) the existence of equilibra in general multiplayer, multi-strategy evolutionary games is investigated. They consider games ranging from the basic two player two strategy game to n-player, d-strategy games. For simple games a 2d payoff matrix is sufficient to represent the game but for the general case of n-player, d-strategy games where the order of players matter they adopt the following tensor notation $\beta_{i_0,i_1,i_2,\ldots,i_{d-1}}$. i_0 represents the strategy of the focal player and the other indexes represent the strategies of the player denoted by their subscripts. The focal player is the player who receives the payoff in question. To represent all possible payoffs for all combinations of players and strategies the number of entires becomes n^d. This is then used as a representation for the "payoff matrix" for a general multiplayer game. Continuing with the analysis of such games they also derive that the number of internal equilibria in such multiplayer games are $(d-1)^{n-1}$.

2.4 Evolutionary optimization algorithms

Evolutionary algorithms is a group of heuristic optimization methods inspired by evolution and natural selection (Whade, 2008). The main steps of a basic evolutionary algorithm is as follows:

1. Generate a population of individuals each representing a point in the solution space
2. Evaluate each individual according to a fitness function capturing the factors to be optimized
3. Select individuals based on fitness to form a new population
4. Introduce random mutations in the new population
5. Redo step 2-4 until convergence or desired optimality is attained

Each step can be refined to various degrees, for example it is possible to initialize the population completely randomly or start with a set of decent solutions. It turns out that as long as there is a slight bias towards selecting individuals with high fitness the algorithm will move to improve the traits selected for. A great strength of evolutionary algorithms is that they are very general to the choice of fitness function. Select a mathematical function as the fitness function and the algorithm will optimize it or set it to the distance moved in a gate simulation environment and the algorithm can evolve walking patterns that do well. The mutation and selection step also offer design choices affecting search pattern and convergence speed. In this project a simple EA is used for global optimization on the Urdar model.

3 The Urdar Model

The Urdar model was was first constructed by Philip Gerlee and Torbjörn Lundh and is covered in great detail in (Gerlee and Lundh, 2010a) and is further examined in (Gerlee and Lundh, 2010b). This section briefly explains the dynamics of the model but also highlights some key aspects that are of special significance to later results.

3.1 The basic properties of the model

The world of Urdar is populated by organisms represented by cellular automaton rules as described in 2.1. The organisms consume binary strings through application of the ca-rule they represent. Energy is extracted corresponding to the change in entropy of the string caused by the rule. Entropy was previously discussed in section 2.2. The entropy is approximated as the Shannon entropy as calculated at level two as seen in equation 5.

$$s = \sum_{\sigma_3} p(\sigma_3) log_2(\frac{1}{p(\sigma_3)}) - \sum_{\sigma_2} p(\sigma 2) log_2(\frac{1}{p(\sigma_2)}). \tag{5}$$

Where σ_3 are the blocks of size three and $p(\sigma_3)$ the probability distribution of the same blocks in the string. σ_2 describes the same but for block size two. The energy is then defined as $E = 1 - s$. Digestions that decrease the entropy are not permitted to occur otherwise the ca-rules would be able to create energy from nothing. Instead a fresh supply of bit-strings are continuously added to the system while a portion of the current strings are removed. Ca-rules that perform well will increase in abundance and ca-rules that do poorly will decrease.

3.1.1 Analogy to a real system

Relating the Urdar model to real systems it most closely resembles a chemostat reactor where there is a constant flux of solution through the system with species of bacteria(or other organisms) living at an apparent steady state inside. Fresh nutrients flow into the system replacing a portion of the nutrients already inside. The rate at which solution is replaced is called the dilution rate. A dilution rate of 0.3 indicates that 30% of the total solution is replaced with fresh solution each time interval. Here lies the main difference from the analogy between a chemostat and Urdar, in a real chemostat the organisms directly affected by the dilution rate as they to are flushed out with the solution. The dilution rate also directly determines the growth rate of organisms as the number of new bacteria needs to be equal the number of bacteria removed from the system in outflowing solution. Instead in Urdar there is no removal of bacteria through flow and the population size is still modeled as constant for all dilution rates.

3.2 Population dynamics

As stated the total population size is constant. The ca-rules compete over the limited space. When an organism successfully digests a string there is a probability based on the energy gain that it will divide. In the current implementation of Urdar this probability function has the form.

$$P(\Delta E) = \begin{cases} \frac{1-exp(-\Delta E \beta)}{1-exp(-\beta)} & \text{if } \Delta E > 0 \\ 0 & \text{if } \Delta E < 0 \ . \end{cases} \tag{6}$$

The new copy will randomly replace any another organism in the population. As the division occurs there is small probability that the new copy will mutate into any of the other 255 other species. There is no complex genotype to phenotype mapping and the mutation procedure is simply generation of a random number on the range [0 255].

A population or ca-rules in Urdar is represented by a vector of length 256. Each entry represents the frequency of the corresponding ca-rule. The sum of all entries must equal one. Denoting the frequency of rule i with x_i the condition becomes $\sum_i x_i = 1$.

3.3 Dynamics of the food-pool

The key aspect that enables cross-feeding and interactions between species in Urdar is the sharing of food. Organisms interact by digesting food strings that has been previously digested by other organisms. Thus the extent of interactions and cross-feeding possible is limited by the food dynamics. At each time step all organism get a chance to eat. They all randomly pick a food string out of the pool and depending on if they succeed to digest it they put it back modified or unmodified. Next a percentage of all strings is replaced with fresh strings. This percentage corresponds to the flow through the system also know as the dilution rate. The result is two opposing processes, first the organisms digests strings increasing the age of strings and second the flow of new strings into the system rejuvenates the food-pool. At high dilution rates strings quickly get replaced resulting in a very young food-pool while low dilution rate allows for many partially strings to accumulate in the system. The previous work Gerlee and Lundh (2010a) showed how the dilution rate affects many aspects of the model.

Fresh food strings are setup in the following manner. First 50% of strings are initialized as all zeros and 50% as all ones both types of length 100. Then 5% of the bits in these strings are randomly flipped to provide digestion entry points for the ca-rules.

3.4 The concept of optimality in urdar

Each individual digestion increases the entropy of the food pool by a tiny bit. By measuring the entropy of strings flowing out of the system it is possible to estimate the ability of the population to digest strings. By comparing this to the entropy flowing into the system and normalizing it is possible to calculate the efficiency as defined in (Gerlee and Lundh, 2010a).

$$\eta = \frac{\sum_k (E_0 - E(r_k^{out}))}{\gamma N_R E_0}. \tag{7}$$

Where η denotes the efficiency, E_0 the mean energy of strings flowing into the system, γ the flow rate, N_R the number of strings in the system and $E(r_k^{out})$ the energy of string k flowing out of the system. This becomes a measure in the range [0 1] that determines the efficiency of the current population. Zero corresponding to strings flowing out of the system undigested and one to strings being fully digested. Even thou a efficiency of one is theoretically possible it is not achievable in the actual simulation. Partly because strings are replaced with new strings before they are digested enough time. But even at very low flow rates it is hard for rules to extract the last bits of energy from the strings.

4 Results and Analysis

This section presents and describes the results of this project. Due to the nature of the project the section contains many varied lines of approach. It is possible to go directly to the subsection of interest but in many cases the later sections expand on earlier results. Overall the ordering of the section follows the order in which the results were generated.

4.1 Global optimization

The first approach to develop understanding of optimality in Urdar was to look at it from a global perspective. More specifically for a given flow rate which is the most efficient population? And also if this theoretically best population appears in the simulation. To answer this the idea was to use a global optimization method on the system.

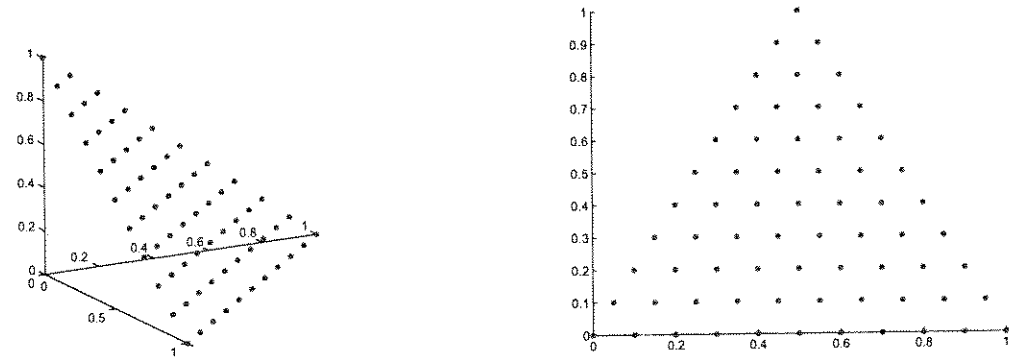
The initial thought was to use a classical method such as gradient decent. There are a few complications which makes this hard to perform however. Since it Urdar is a computer model there is no mathematical function to optimize instead measurements are taken on the system. The gradient is not easily defined and would have to be approximated through repeated measurements. This would have to be done in each of the 256 dimensions. Instead heuristic methods were investigated as a possible way to find the optimal population.

For the first investigation of this an evolutionary algorithm (see 2.4) was chosen. This was implemented in the following way. Fitness was defined as the average efficiency of the system after a simulation over 300 iterations. An individual in the EA was defined as a vector of length 256 representing a population of rules in Urdar. The starting set of solutions was generated randomly distributing lumps of size 0.05 over the rules making sure the final populations summed to one. A lower limit to rule fractions when initializing populations was set to avoid seeding populations which are too diverse in their setup. Mutation was implemented as simply taking a random small fraction from one rule and adding it to another. The starting populations were in general quite bad but running the algorithm quickly finds populations with levels of efficiency comparable to the ones in actual the actual simulation. The process then slowed down considerably and no populations that were exceptionally good were identified. The key observation here was that several diverse populations were able to reach levels of efficiency comparable to the highest observed in both simulation and optimization. This indicates that the optimality surface is quite shallow. It also raises the question if a system moves towards a high level of efficiency but not towards the globally highest level of efficiency is efficiency still optimized for?

Given that a very good population is found using a global optimization method there is still uncertainty if the population is the best possible. Another approach to examining optimality is to consider the upper limit in terms of efficiency that the dynamics can achieve. This was not done however due to time limitations but left to future work see section 5.1. Instead the focus was turned back down to simplified dynamics with only a few rules interacting at a time.

4.2 Reduced urdar dynamics

Turning away from the global aspects of optimality it is possible to get a more transparent picture by examining the behavior of only a few rules at a time. Previously in (Gerlee and Lundh, 2010a) pairwise invasion experiments were conducted to examine, among other things, which rules can coexist. An extension to this was done in (Gerlee and Lundh, 2010b) where the goal was to find triplets of rules exhibiting rock, paper, scissor dynamics. The search was done in the subset of rules that, based on the invasion experiments upheld the following. Rule A can invade rule B which in turn can invade rule C which finally can invade rule A. Unfortunately when all three rules were mixed together the result was that one of the rules quickly went extinct and the stronger of the remaining two took over the population.



(a) Sampled populations in 3D

(b) Sampled populations mapped to 2D

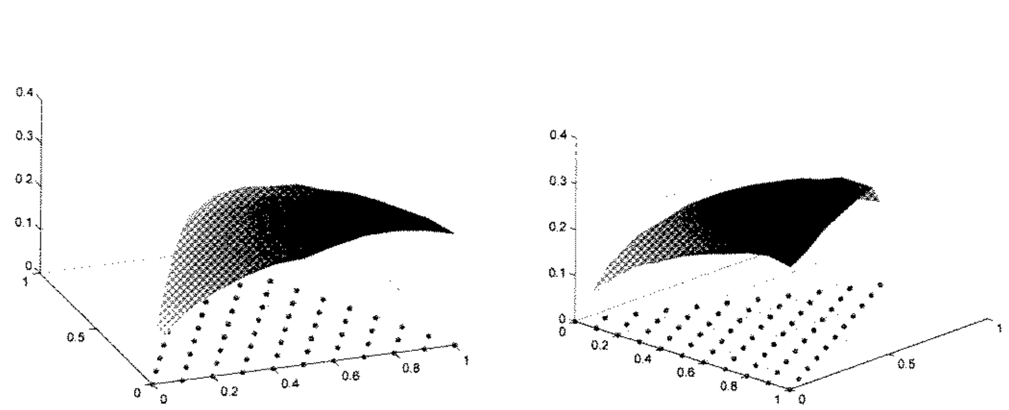
Figure 2: Example of how the mapping of three rule populations onto a 2D-simplex is performed. Each point represents a population of three rules. Panel a show how these form a surface in three dimensions. Panel b show the same points as in a but mapped to the 2D-simplex according to equation 8

Instead of looking for triplets of rules that exhibit RPS-dynamics and cycling it is possible to look for triplets of rules that simply are able to coexist. The dynamics will not be that of limit cycles but instead that of simple stable fixed points. There exists a nice way to visualize populations of three species. Three dimensions are needed to describe the fractions of each rule but since the sum of all frequencies must equal to one all the points lie on a triangle on a plane in three dimensions as seen in figure 2a. It is possible to map this simplex down to two dimensions allowing according to equation 8 yielding the result seen in figure 2b.

$$\begin{aligned} x_2 &= 0.5 - \tfrac{x_1}{2} + \tfrac{y_1}{2} \, , \\ y_2 &= z_1 \, . \end{aligned} \tag{8}$$

Over this simplex a response surface can be used to visualize the productivity or any other quantity of interest. Two examples of triplets visualized in this manner can be seen in figure 3. Both examples have in common that there is an observable drop in efficiency at the sides meaning that situations where the three rules coexists are generally more productive than those of the nearby two-rule populations. Indicating that there is some correlation between rules being more productive together and being able to live together.

By measure how the dynamics in Urdar shift the population on the simplex it is possible to visualize fixed points in the three rule dynamic. This is done by initializing the population at one of the sampling points on the simplex, iterating the dynamics for a few steps, repeating this procedure several times and measuring the average direction in which the population moves. This generates an approximate value for the local derivative which can by visualized in the from of a vector field. This was



(a) Efficiency surface for rules 99, 109 and 120 (b) Efficiency surface for rules 98, 105 and 165

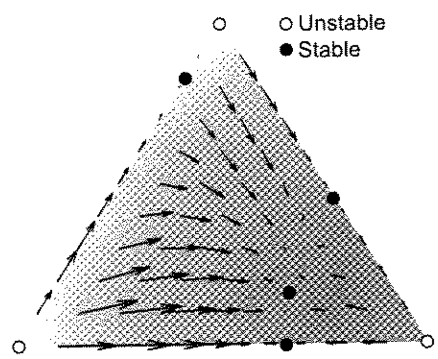
Figure 3: Examples of efficiency surfaces for two rule triplets. Generated by measuring the efficiency in the points seen in figure 2a. The height and the color shade both represent efficiency. The higher points and the darker shade of gray the more efficient is the population at the point. The efficiency was measured at flow 0.03.

done for both the above cases and the result can be seen in figure 4. It shows two cases with similar behavior. A stable fixed point indicates that a population initialized on the interior of the simplex will reach this fixed point and stay there. Note that Urdar is still a stochastic process and as such these are not true fixed points and that the population will fluctuate about them. There is a always chance albeit small that the population will drift so far as to collapse to one of the edges. Populations initialized at an unstable fixed point will remain there indefinitely until unless they get a slight perturbation in which case they will move towards the nearest stable fixed point. This is only true if mutations between rules are excluded which is the case in these realizations. Without mutation such perturbations do not occur. With mutations included they would constantly occur and no population could remain at an unstable fixed point for long.

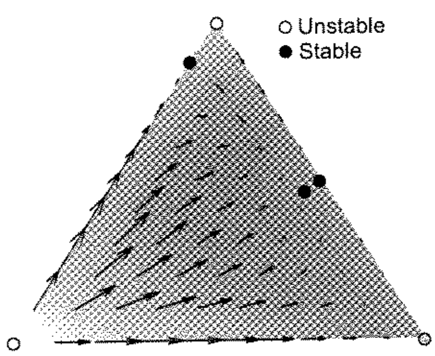
The exclusion of mutations also mean that the dynamics on the edges of the simplex are separate from the interior dynamics and represent pairwise interactions between rules. It is possible to move from the interior to one of the edges but it is not possible to move back. This is important to note and means that there is in fact four different interactions shown in each simplex. The interior dynamics between all three rules and the dynamics on the edges between the three different pairs of rules. The stable fixed points on the edges only exist for the two player dynamics on that edge. There is a special case for the bottom right corner in figure 4b. Each corner actually represent two fixed points, one for each two player interaction that includes that corner. The corner in question represent a population of 100% rule 105. In the interaction between 105 and 98 along the bottom edge this is a stable fixed point. In the interaction between 105 and 165 this is an unstable fixed point.

4.3 Measuring interaction strength between rules

Up until now the approach has been to run the full Urdar model taking measurements with the aim of understanding the forces driving the system. This method drawbacks in that it is hard to determine the cause for observed behavior, how different effects intermingle and how large the effect of noise is. Instead it is possible to throw away big parts of the model. The interest currently lies in understanding optimality in



(a) Dynamics on the simplex for rules 99, 109 and 120



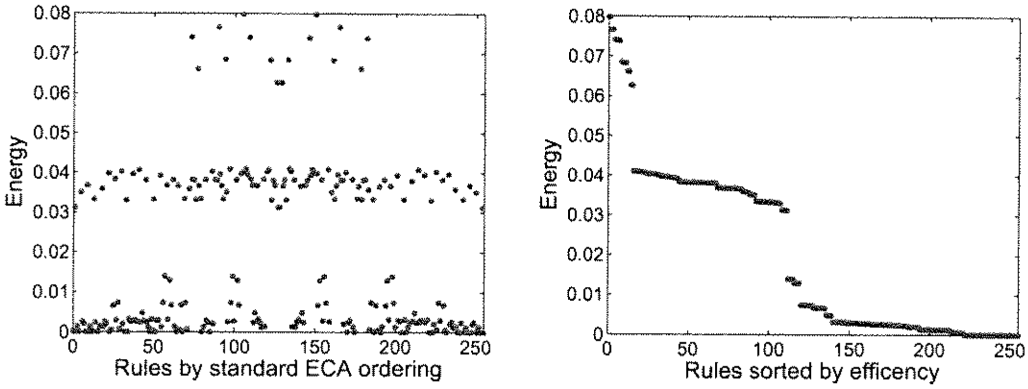
(b) Dynamics on the simplex for rules 98, 105 and 165

Figure 4: Visualization of the population dynamics on the simplexes. The vectors indicate how the populations moves over the simplex. The length of the vectors correspond to how fast the population changes. Note that mutations between rules are not included in this realization. This means that once the dynamics lands on an edge or in a corner meaning that a rule goes extinct that rule cannot reestablish itself in the population. The darker shade of gray on the surface indicate higher efficiency for a population. Note that the dynamics moves around randomly in an area around the fixed points which means that the position of the stable fixed points is approximate.

Urdar and which rules are the most efficient. To measure this the population and evolution dynamics can be disregarded and focus should be put on the food dynamics. Instead of observing how well a rule does in the simulation we take that rule out of the simulation and measure its ability to extract energy from food strings. The benefit is twofold as the energy extracted by a rule is directly related to its own fitness but also information is gained regarding which rules work well together by letting rules digest strings previously digested by other rules.

Remember that the key element in Urdar is cross-feeding, that food strings are digested in many steps by different rules. When rule A digests a string previously digested by rule B the energy extracted can be very different than if rule C instead of B had digested the string first. The notation chosen to describe these interaction terms is as follows. First define E_0 as the mean energy in fresh strings. Then define E_i as the mean energy left in strings digested by rule i and E_{ij} as the mean energy left in strings digested first by rule i then rule j. The first order interaction describing rules eating fresh strings is then defined as: $w_i = E_0 - E_i$, for rules consuming strings previously digested once $w_{ij} = E_i - E_{ij}$. In a similar way third order interactions are defined as $w_{ijk} = E_{ij} - E_{ijk}$. It is important that the energies are averaged. The interaction terms are intended to describe the simulations and in the simulation the food strings are initialized as bit strings of ones or zeros with length 100 and 5% of bits flipped. This means that there is a small variation to the energy extracted depending on the placement of the flipped bits and a large variation caused by 50% of strings initialized as zeros and 50% as ones as some rules are only able to handle one or the other.

The digestion measurements on fresh strings are taken by simply applying the ca-rule in question to the bit string and calculating the difference in energy. Note that the restriction that entropy cannot decrease still holds, i.e. a rule cannot increase the energy left in a string, and transformations that cause the entropy to decrease are counted as zero energy gain. For longer interactions the preceding ca rules are first applied to the bit string and the interaction coefficient is then measured as the energy the final rule can extract. Here the concept of decreasing entropy becomes even more important. In order to properly separate the interaction coefficients, situations where the intention is to measure E_{ijk} but j or i fails to digest the string that realization is set to zero even though k might extract energy in the end. The intention is to let w_{ijk} describe exactly the average energy when rule k digests a string previously digested by i and j. When i or j fails to digest the actual energy becomes E_{jk} or E_{ik} which should not be included in the mean. The trials are still included in the mean for w_{ijk} but as zeros, since it should reflect the interaction strength between the rules. If rule k is able to extract high energy from strings previously digested by i and j but only manages this once in ten trials the interaction is still fairly weak.



(a) First interaction term w_i for all rules

(b) First interaction term w_i for all rules

Figure 5: Both the figures show the same data but ordered in different ways to give a feeling for the distribution of strengths. From a it is visible that pairs of mirrored rules are equally good at digesting strings. The best pair visible in a is 105 and 150. Also worth mentioning is the second best pair which is 90 and 165. Make a mental note that the top energy extracted is about 0.08 when digesting fresh strings.

Making the actual measurements takes considerable time. Single step interactions, w_i take less than a minute to complete for all rules. But for each additional interaction step we want to calculate the number of combinations increase with a factor of 256. In figure 5 the first order interaction terms can be seen for all rules. The division into three visible regions at ≈ 0.8 , ≈ 0.4 and ≈ 0 is likely caused by the chosen structure for fresh food where 50% are mostly zeros and the rest mostly ones. The three regimes would then be rules that can digest almost all strings, rules that can digest either black or white and rules that cant digest either. Figure 6 is an attempt to visualize all the second order interaction terms (w_{ij}). The aim of the figure is not to give an instant view of what rules have the strongest interactions but rather to give a notion of how the interaction space looks. The darker the pixel the higher the interaction value. The figure is not symmetric when reflected top to bottom but is so from right to left in the same way as in figure 5a. It is also possible to see similarities in 5a and figure 6 in that rules that are weak in the first digestion are also weak in the second. The white stripes across both directions in figure 6 are caused by such weak rules. The final thing to note, which is not possible to read from the figure, is that largest second order terms are as high as 0.14 compared to 0.8 for first order terms.

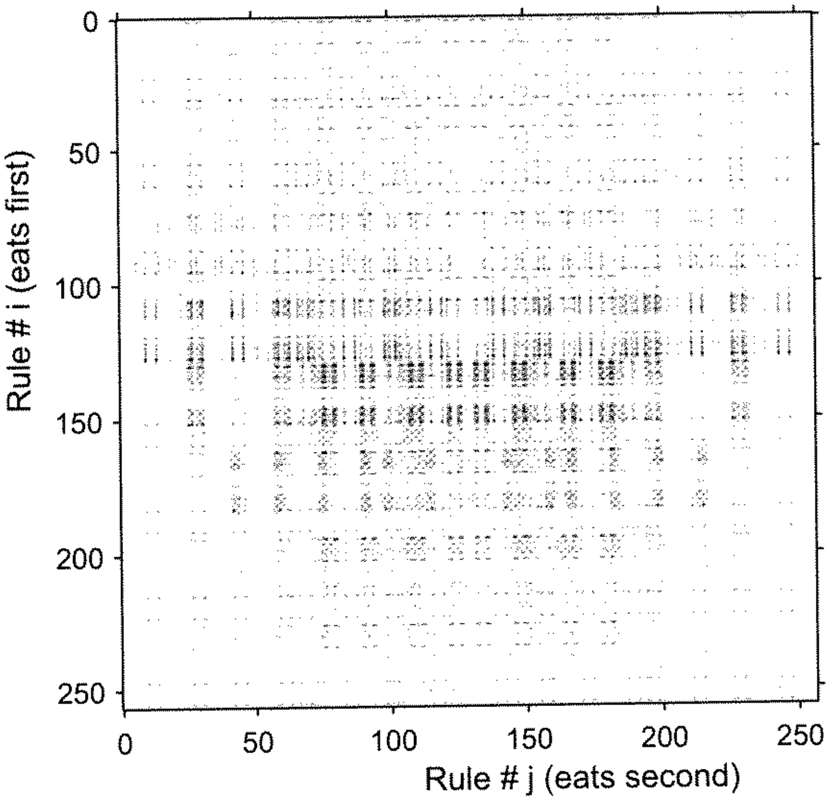


Figure 6: Second order interaction terms, w_{ij} measured for all rules. Dark pixels correspond to strong interactions i.e. high energy extracted by the second rule and white are very weak interactions. Shows similarities in structure to the first order interactions shown in figure 5a. Rules that are poor at digesting fresh strings are bad partners to interact with and that the strongest interactions are still found in the center region. Note, even thou it's not visible in the figure, that the highest second order interaction value is 0.14.

4.3.1 A first attempt at approximating interaction strengths

As stated previously the computational cost of calculating higher order interaction terms scales very badly. The number of possible combinations at for interactions of length n is 256^n. It is conceivable however that higher order terms might be approximated from lower order terms. Torbjörn Lundh proposed the following as a first attempt at doing this. In addition to the notation used above lets introduce the ratios between energy before and after digestion as $K_i = \frac{E_i}{E_0}$ and $K_{ij} = \frac{E_{ij}}{E_i}$. Now the assumption is that regardless of previous digestion histories relative energy uptake of a rule is roughly equal. That would mean that $K_{ij} \approx K_j$. Given some algebra this allows for a way to approximate higher order terms from lower order terms:

$$w_{ij} \approx \frac{E_0 - w_i}{E_0} w_j. \tag{9}$$

And for third order terms:

$$w_{ijk} \approx \frac{E_0 - w_i}{E_0} \frac{E_0 - w_j}{E_0} w_k. \tag{10}$$

The assumption has some merit but it turns out however that this approximation does not hold well. There are a few reasons for this. The first is that energy gain is not relative and can in fact be higher for the second digestion than for the first as show previously in figure 5 and 6. Figure 7 show the time evolution of ca-rule 90 and the corresponding entropy for each time step. The entropy differences between time steps two, one and zero do follow the assumption that extracted energy is relative to the energy left but after step two the assumption fails completely. This is an effect of how entropy is calculated and is non-obvious in terms of what patterns are intuitively expected to have high entropy. The key is that the entropy is calculated as the difference in block entropy between block size three and block size two as seen in equation 5. If a pattern exhibit an even mix of block types at level three and an even mix of patterns at block size two the approximated Shannon entropy is zero. Patterns of ...01010101... show exactly this behaviour with an even mix of 01 and 10 at size two and 010 and 101 at size three. This is why ca-rule 90 is not able to increase the entropy at time step three in figure 7.

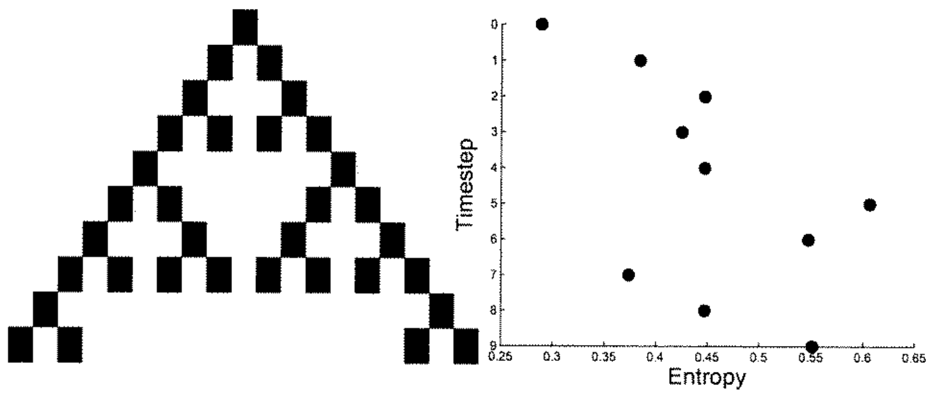


Figure 7: On the left are the first few time steps for rule 90 when starting from a single one history. On the right is the Shannon entropy approximated at level two for each time step. The key thing to note is that the entropy does not increase constantly over time instead there are plateaus. Since there is a restriction that a rule in Urdar is not allowed to decrease the entropy of a string rule 90 cannot fully digest this string alone. This decrease in entropy visible at step 3, 6 and 7 is caused by patterns that although containing an even mix of ones and zeros have roughly the same entropy at block size two and block size 3 thus having a low approximated Shannon entropy.

Nevertheless approximations of higher order terms might be fruitful and even necessary due to computational costs. Another approach would be to use second order interactions, w_{ij} etc, as a basis for the approximation. Even such a scheme might also fail however considering the example in figure 7. The rule performs well for two steps but is unable to digest the string again. This behavior would be hard to capture starting from w_{ij} terms. The approximation based on second order terms has not been tested however and is left to future work.

4.4 Urdar formulated as the replicator equation

With a framework to handle the how well rules perform together and alone many new paths of investigation open up. Philip Gerlee proposed the following formulation of a fitness function for ca rules in Urdar:

$$\phi_i = F(0)w_i + F(1)\sum_j x_j w_{ji} + F(2)\sum_k \sum_j x_j x_k w_{kji} + ... \quad (11)$$

$F(0)$ describes the fraction of fresh strings in the food pool, $F(1)$ describes the fraction of strings digested once(for more details see section 4.4.1). w_i, w_{ji} and w_{kji} are the interaction coefficients as described in section 4.3. x_j is the fraction of rule j in the population. Terms for longer interactions are included in a similar way.

The equation describes the fitness of rule i as a function of it's ability to digest fresh strings, it's ability to digest strings previously digested once weighed by the fraction of rules in the population and similar for longer interactions. Each term is also weighed by the fraction of strings of a certain age in the population. This distribution over string ages is of course strongly dependent on the dilution rate as described in section 4.4.1.

There are a few assumptions that might not be obvious that underlie this description of the fitness. One assumption that isn't clearly stated is that dynamics of the food pool must work on a shorter timescale than the population dynamics. What this means is that if the population shifts the strings in the food pool must be quickly replaced with fresh strings that are then digested to make the digestion histories of strings in the pool properly reflect the population structure. In the above formulation it would also appear that the string age distribution is constant regardless of population structure which is far from true when considering all populations. A population of weak rules(for example the ones with low score in fig ref) will have a much fresher food pool as they are often unable to digest the strings. In the subset of strong populations the assumption still holds however and since these are the ones of interest the assumption is reasonable.

The fitness function defined in 11 can now be used in a replicator equation to provide a way to generate the full population dynamics. The replicator equation defines the rate of change for a rule to be proportional the current fraction of that rule times it's fitness relative to the average fitness in the population.

$$\frac{dx_i}{dt} = x_i(\phi_i - \overline{\phi}). \tag{12}$$

The average fitness is defined as:

$$\overline{\phi} = \sum_j \phi_j x_j. \tag{13}$$

A rule that has higher fitness than the average fitness will increase it's fraction of the population while a rule with less than average fitness will decrease. Note the above formulation mutations between rules is not included.

4.4.1 Age distribution for food strings

In order to assign values to the $F(\cdot)$ weights in the previously described fitness function (ref first replicator) we need to know the age distribution for food strings. The age of a string corresponds to the number of times it has been digested. The dynamics of the food pool was described in section 3.3. As previously mentioned there are two competing processes that affect string age; strings being eaten and the flow of strings through the system. These processes can be treated analytically. First we note that in the current version of Urdar the number of strings in the food-pool is 5000 and the number of organisms is 1024. This gives a ratio of approximately $\frac{1}{5}$ that can be digested each time step. Assuming that all digestions succeed and that the distribution has reached a steady state the fraction of strings of a specific age can be described in the following way.

$$F(n) = \frac{1}{5}F(n-1) + F(n) - \frac{1}{5}F(n) - \gamma F(n). \tag{14}$$

The equation describes the fraction of strings of age n at time t as a function of string fractions at time t-1 but since steady state is assumed the time dependence is excluded. In order the right hand side terms describe: strings of age n-1 that get digested, strings of age n that stays, strings of age n that get digested and strings of age n that flows out of the system. The equation can be simplified into:

$$F(n) = \frac{F(n-1)}{1+5\gamma}. \tag{15}$$

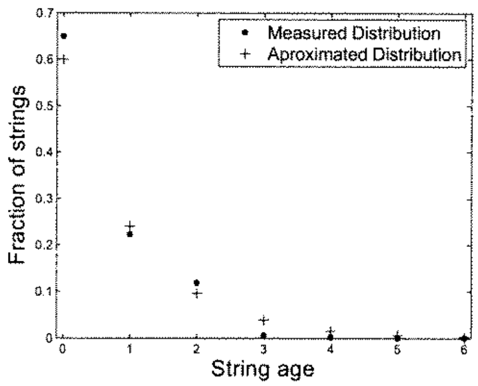
Now noting the special case of fresh strings, $n = 0$, derived from $F(0) = F(0) - \frac{1}{5}F(0) + \gamma(1 - F(0))$:

$$F(0) = \frac{5\gamma}{1+5\gamma}. \tag{16}$$

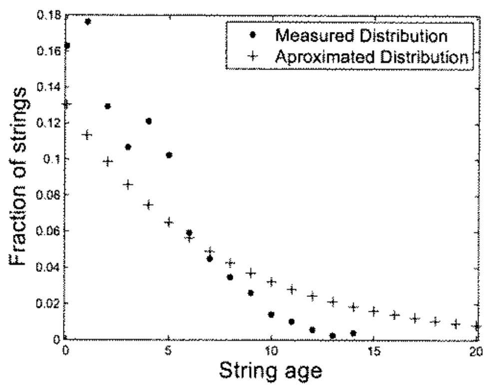
Going back 15 and inserting 16 it is possible to form the general expression:

$$F(n) = \frac{5\gamma}{(1+5\gamma)^{n+1}}. \tag{17}$$

In figure 8 a comparison of the distribution of ages in the food-pool between what



(a) Distribution of string ages for flow 0.3



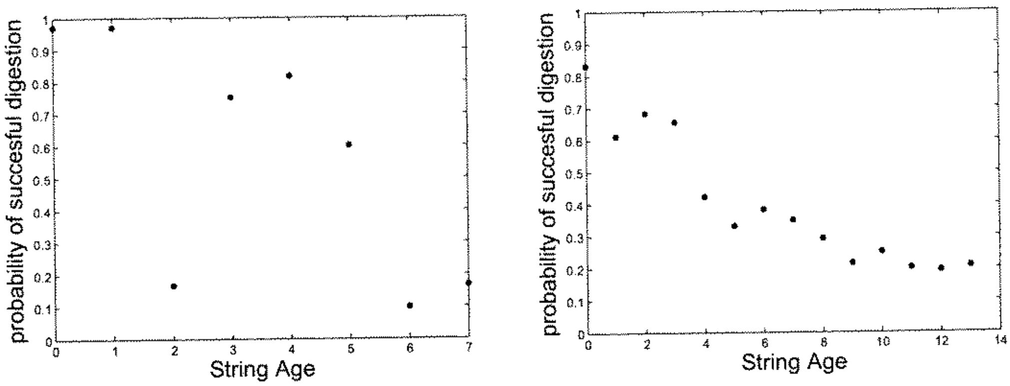
(b) Distribution of string ages for flow 0.03

Figure 8: The measured and approximated distributions over string ages. Note the difference in scale for the two examples. For flow 0.3 in 8a there is a large amount of fresh strings and the approximation fits the measured data well. For flow 0.03 8b the approximated distribution does not fit the measurements as well and the strings are fresher than expected. The main reason for this is that the probability for digestion is not 100% and not uniform over string ages. On the contrary it drops rapidly as seen in figure 9. This lowers the rate at which strings are digested which in turns means that fewer strings achieve a high age before being flushed out.

is generated by the expression in 17 and what is measured in an actual simulation can be seen. The cause for the difference between the measured distribution and the approximated lies in the assumption that all digestions succeed.

4.4.2 Game theoretic interpretation

The above reformulation of the Urdar dynamics can be seen as a game theoretic formulation of the interactions. In Urdar the strategies correspond to rules and the number of players are the interaction length. Essentially we have a population of strategies or rules simultaneously partaking in games with different number of players. The digestion of a fresh string corresponds to a trivial one player game where a rule simply receives payoff based on itself. Digestions of strings previously digested once become two player games and three digestions become three player



(a) Probability for successful digestions at flow 0.3 (b) Probability for successful digestions at 0.03

Figure 9: The probability of successful digestions for strings of different ages. Neither of the two cases seen correspond well to the approximation that all digestions are successful. Still for the case of high flow and strings of age zero and one the approximation fit well and as a result the the calculated and measured data in figure 8a to be very similar.

games etc. This distribution over games of different number of players is determined by the flow rate as described in section 4.4.1. The tensor notation described in section 2.3.1 now becomes equivalent to the one used describe interaction terms with a slight adjustment to formalism. While the tensor notation used in (Gokhale and Traulsen, 2010), $\beta_{i_0,i_1,i_2,\ldots,i_{d-1}}$, put the focal strategy first the notation used for interaction terms, $w_{ijk\ldots l}$, lets the last index notate the strategy receiving the payoff.

4.4.3 Replicator equation for two rules

Consider a population consisting of only two rules A and B their fractions denoted by x and y. Since the total population sums to one and the part that isn't x is y it is possible to reformulate $y = 1 - x$. Given this it is possible to reformulate the replicator equation in 12 into:

$$\frac{dx}{dt} = x(1-x)(\phi_A - \phi_B) = P(x). \tag{18}$$

Leaving a single differential equation to determine the system. Now arguing that we want to examine a system of two rules at high dilution rates it is possible to throw away interactions of length three and higher. It is then possible to write the fitness functions for A and B as:

$$\phi_A = F(0)w_A + F(1)(w_{BA}(1-x) + w_{AA}x), \tag{19}$$

$$\phi_B = F(0)w_B + F(1)(w_{AB}x + w_{BB}(1-x)). \tag{20}$$

This setup allows for analytical treatment of the system. That is locating and characterizing fixed points and examining productivity which among other things allows for prediction on which rules can exhibit stable cooperation in the actual simulation. First noting that a fixed point is defined as point in the state space where the rate of change is zero $\frac{dx}{dt} = 0$. Looking at equation 18 it has two trivial fixed points at $x = 1$ and $x = 0$ corresponding to the entire population consisting of 100% A or 100% B. There is also a fixed point at x^* when:

$$\phi_A = \phi_B. \tag{21}$$

The position of this fixed point is important to predict the dynamics of any given rule pair. If this fixed point lies withing the range $x^* \in (0\ 1)$ and is stable it means that the two rules can coexist. But leaving stability for later lets first look at the position of this fixed point. By expanding equation 21 and solving it for x the following expression for the position of the fixed point is derived.

$$x^* = \frac{\frac{F(0)}{F(1)}(w_B - w_A) + w_{BB} - w_{BA}}{w_{AA} + w_{BB} - w_{AB} - w_{BA}}. \tag{22}$$

It is possible to devise conditions for when the fixed point lies within $x^* \in (01)$ but first consider the concept of stability for fixed points. Consider the situation where system resides at the fixed point. The system now receives a small perturbation and is pushed away from the fixed point. If the dynamics drives the system further away from the fixed point it is considered unstable. If instead it is drawn back to the fixed point it is stable. In more formal terms the stability is investigated by linear stability analysis which corresponds expanding the set of differential equations to the first order at the fixed point. This yields a stability matrix and the eigenvalues of this matrix determines the stability (Weissten, 2011). For this one dimensional case however this the stability matrix simply becomes a scalar value corresponding to the derivative of equation 18 and the sign of this value will determine the stability. If positive any perturbation will grow exponentially over time corresponding to an unstable fixed point and if negative perturbations will decrease exponentially corresponding to a stable fixed point. Calculating the time derivative of equation 18 yields:

$$P'(x) = (1-x)(\phi_A - \phi_B) - x(\phi_A - \phi_B) + x(1-x)(\phi'_A - \phi'_B). \tag{23}$$

Given that equation 21 holds this reduces to:

$$P'(x^*) = x(1-x)(\phi'_A - \phi'_B). \tag{24}$$

Noting that $x \in (01)$ makes sure that $x(1-x) > 0$ thus the only term influencing the sign of $P'(x)$ is the derivatives of the fitness functions.

$$\phi'_A - \phi'_B = F(1)(w_{AA} - w_{BA}) - F(1)(w_{AB} - w_{BB}).$$

Noting that $F(1)$ is a positive constant the expression above is negative if the following holds.

$$w_{AB} + w_{BA} < w_{AA} + w_{BB}. \tag{25}$$

This equation captures the stability criteria for the internal fixed point. It has an intuitive interpretation. If the two rules gains more from eating each others leftovers than they do from eating their own leftovers the internal fixed point will be stable. That is to say they are able to live together granted that the fixed point is in fact internal. Moving back to equation 22 it is possible to state criteria for when the fixed point is internal i.e. $0 < x^* < 1$.

$$0 < \frac{\frac{F(0)}{F(1)}(w_B - w_A) + w_{BB} - w_{BA}}{w_{AA} + w_{BB} - w_{AB} - w_{BA}} < 1.$$

If this is to be simplified the sign of the denominator $w_{AA} + w_{BB} - w_{AB} - w_{BA}$ needs to be determined. Conveniently this corresponds to the stability criteria for the fixed point. When looking for the stable internal fixed points the denominator is negative and for the unstable fixed points the denominator is positive. This allows the following separation of the two cases. For stable internal fixed points the criteria becomes.

$$F(1)(w_{AA} - w_{AB}) < F(0)(w_B - w_A) < F(1)(w_{BA} - w_{BB}). \quad (26)$$

And for the internal but unstable fixed points.

$$F(1)(w_{AA} - w_{AB}) > F(0)(w_B - w_A) > F(1)(w_{BA} - w_{BB}). \quad (27)$$

These criteria also contain some intuitive interpretations regarding the existence of the internal fixed point. The left hand side describes the relationship between A consuming A's leftovers and B consuming A's leftovers. If the w_{AB} term is much larger than the w_{AA} term the lhs will be small. This would increase the range of possible values for the center expression that give a stable fixed point but decrease them for the unstable fixed point. That would agree with the notion that large cross-feeding terms promote coexistence. The same argument holds for the right hand side. A large w_{BA} term compared to the w_{BB} term promotes stability.

There is still two fixed points that have not been characterized for this simplified system namely $x = 0$ and $x = 1$. The stability of these fixed points is also determined by the sign of $P'(x)$. For $x = 0$ equation 23 reduces to.

$$P'(0) = (\phi_A - \phi_B).$$

And for $x = 1$ it reduces to.

$$P'(1) = -(\phi_A - \phi_B).$$

Inserting $x = 0$ and $x = 1$ into equations 19 and 20 which describes the fitness functions reduces the above to.

$$P'(0) = F(0)(w_A - w_B) - F(1)(w_{BB} - w_{BA}).$$

And for $x = 1$ it reduces to.

$$P'(1) = F(0)(w_B - w_A) - F(1)(w_{AA} - w_{AB}).$$

Constraining the above expressions to be negative we arrive at criteria for when $x = 0$ and $x = 1$ are stable fixed points. For $x = 0$:

$$F(0)(w_A - w_B) < F(1)(w_{BB} - w_{BA}). \quad (28)$$

For $x = 1$:

$$F(0)(w_B - w_A) < F(1)(w_{AA} - w_{AB}). \quad (29)$$

Again the resultant criteria are intuitively understandable. If for example both w_B and w_{BB} are large rule B is both good at eating fresh strings and it's leftovers. This would also indicate according to equation 28 that $x = 0$, corresponding to a population of 100% B, is a stable fixed point which is the expected result. The same reasonings holds for the fixed point at $x = 1$ with focus on rule A.

Note that the criteria for endpoint stability are not mutually exclusive. Both endpoints can be stable or unstable at the same time. This corresponds to the situation where there is an internal fixed point and it is unstable or stable. It turns out that the criterion for unstable internal fixed point and criterion that both endpoints are stable are exactly equivalent. By rewriting equation 28 into

$$F(0)(w_B - w_A) > F(1)(w_{BA} - w_{BB}).$$

and combining this with equation 29 the following is attained.

$$F(1)(w_{AA} - w_{AB}) > F(0)(w_B - w_A) > F(1)(w_{BA} - w_{BB}).$$

This is exactly the condition for existence of an unstable internal fixed point. The same holds when looking at an internal stable fixed point and both endpoints unstable. This result is no surprise however as in a continuous system it is impossible to have two stable or two unstable fixed points next to each other. Consider the case with two stable fixed points next to each other. Standing in between the two you get pulled into one or the other depending on the distance. However as the force driving the system is continuous there must exist a point in between the two fixed points where their pull is exactly equal. If standing at this point you would remain there indefinitely but take a small step to either side and one of the stable points would pull you in. Which is exactly the definition of an unstable fixed point. Going back to the criteria for the fixed points in the two rule replicator model it is expected that the criterion for an internal stable fixed point is equivalent to the criterion that both endpoints are unstable.

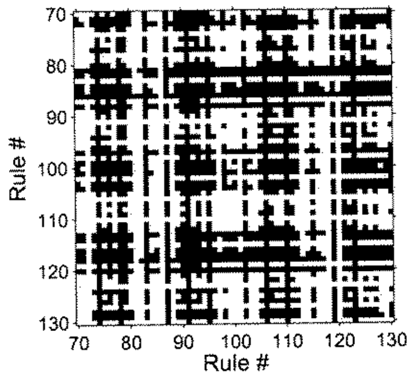
Using the measured data for first and second order interaction shown in figures terms it is possible to apply the derived criteria to predict pairs of rules exhibiting the different kinds of fixed points.

4.4.4 Characterization of two rule dynamics

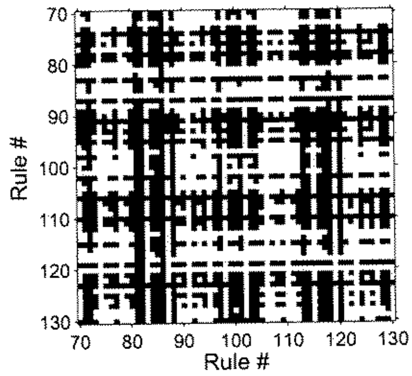
Based on the stability and localization of fixed points the dynamics for two rules can belong to one of four different categories exemplified in (Altrock and Traulsen, 2009). Dominance means that one rule is always the winner regardless of starting position corresponding one stable and one unstable endpoint. Coexistence allows the two rules to live together and is characterized by a stable internal fixed point. Coordination games takes place when there is an unstable internal fixed point and the outcome depends on the starting condition. There is also a situation not mentioned in (Altrock and Traulsen, 2009) and that is the neutral situation with two rules or strategies being equally good. The situation of neutrality is not natural in the circumstances. It would in theory occur between pairs rules whose mappings are inverted for example 90 and 165. These are equally good at extracting energy from strings as discussed in section 2.1. Two things complicate this however. As the Urdar simulation is a stochastic process there is always drift in the population and two equally good rules can still diverge. The second issue is that the interaction terms used in the replicator formulation are measured averages and unless they are rounded to a fixed precision it wont be able to capture true neutrality. Table 1 show the number pairs for each of the four types as predicted by the two rule two player formulation of the replicator dynamics.

4.4.5 Rate difference between replicator representation and simulated dynamics

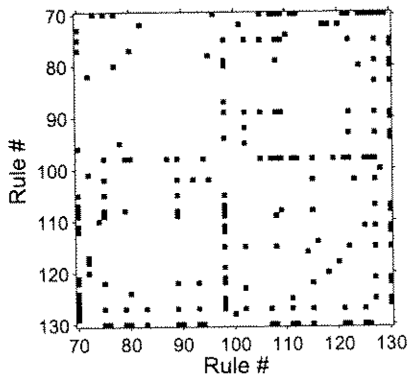
In the replicator dynamics described above the ability to extract energy is taken directly as the fitness. In the simulation however the energy is used as an input to a probability function as shown in equation 6 in section 3.2. The fitness in the simulation instead of being simply energy extracted follows this equation. With $\beta = 10$ this translates to a rate difference when comparing simulation with replicator



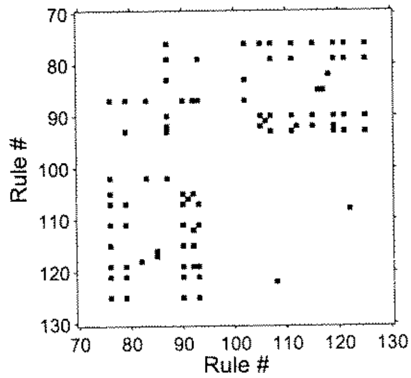
(a) Rule pairs predicted to have x = 0 as a stable fixed point



(b) Rule pairs predicted to have x = 1 as a stable fixed point



(c) Rule pairs predicted to have a stable internal fixed point



(d) Rule pairs predicted to have an unstable internal fixed point

Figure 10: Four plots showing the predicted distribution of fixed points for a subset of all pairs. Black pixels indicate pairs with a stable fixed point except in panel d. Interesting to note is the overlap between distributions. For example a pair with stable fixed points in both panel a and b will also exhibit an unstable fixed point in panel d. Pairs on the diagonal corresponding to picking the same rule twice are trivial.

dynamics. One way to account for this is to approximate equation 6 for small ΔE by Taylor expansion about zero.

$$\frac{1 - e^{-\Delta E \beta}}{1 - e^{-\beta}} \approx \frac{1 - (1 - \beta \Delta E + ...)}{1 - e^{-\beta}}.$$

Including only the first order term and noting that e^{-10} is a very small number equation 6 reduces to.

$$P(\Delta E) \approx \beta \Delta E. \tag{30}$$

However since $\beta = 10$ and ΔE often is as large as 0.1 this is a rather poor approximation. However as a first attempt at accounting for the rate differences that will exemplified in the following section this is sufficient.

4.5 Test of the replicator dynamics

The replicator dynamics has been presented as a way to reduce the Urdar system to a more manageable analytical formulation. It is possible to examine requirements for

Type	Number of pairs
Dominance	30304
Coexistence	1187
Coordination	1139

Table 1: The number of pairs exhibiting the four interaction types. The total number of pairs is $\frac{256^2-256}{2} = 32640$ not counting duplicates and rules paired with themselves. 128 of the pairs above should in reality fall under the neutrality category but they can not be captured by the current measurements of interaction terms. Summing the classes together gives a number 10 short of the expected total. This is caused by a few number of rules who have $w_i = 0$ which cause the replicator notation to collapse to zero. Pairs of of these rules can be considered to belong to the neutral category.

coexistence and predict behavior. Given accurate measurements for interaction terms it is possible to generate the full system dynamics by solving the replicator equation. As only the first two interaction terms have been measured it is not possible to test higher order dynamics. Still the Urdar system at high flow rates is driven mainly by the digestion of fresh and one digestion old strings. This makes it possible to use the replicator formulation to as an approximation to the Urdar system at high flow rates. The following subsections deal with testing on how well the replicator equation capture the simulated dynamics.

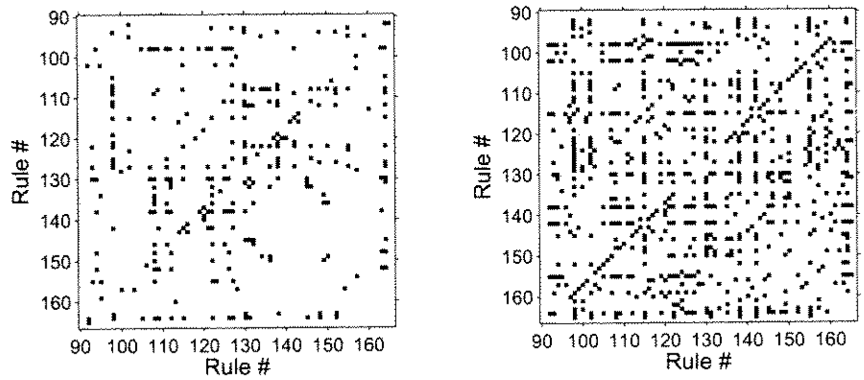
4.5.1 Predicted fixed points for two rules

The replicator formulation of Urdar with two players and two rules show in section 4.4.3 was used to predict rules with stable internal fixed points. It is possible to test these predictions through simulation of the pairs. Letting the two rules live together for many iterations and then checking whether both still are present in the population. This was done for a subset of all possible pairs namely rules in the range 90-165. The result of this testing can be seen in table 2. A visual comparison between predicted and tested fixed points can be seen in figure 11.

Type	Predicted	After 4000 iterations	After 8000 iterations
Stable fixed point	141	121	115
Stable, tested as unstable		20	26
Unstable fixed point	2894	2769	2894
Unstable, tested as stable		270	239

Table 2: Summary of the result when testing fixed point stability. The tests were conducted by simulating the populations for the given time period and checking if both rules survived. The total number of fixed points found to be stable is 391 and 354 for the two time spans is much higher than the predicted number of fixed points. Increasing the number of iterations decreases the number of stable fixed points.

There are a number of possible causes for a population with a predicted stable fixed point to collapse or a population predicted to be unstable to survive. The first cause is the higher order terms that have been excluded in this first representation. Two rules who do not benefit enough from cross-feeding strings that have been digested once can plausibly find more opportunities for cross-feeding on older strings. In a similar fashion populations on the verge of coexistence could collapse due to one rule being able to consistently re-digest it's own waste.



(a) Rule pairs predicted to have a stable internal fixed point

(b) Rule pairs observed to have a stable internal fixed point

Figure 11: Rules exhibiting stable internal fixed points as predicted and tested. Black pixels correspond to rules with stable fixed points. Main thing to note is the great increase in fixed points found when testing.

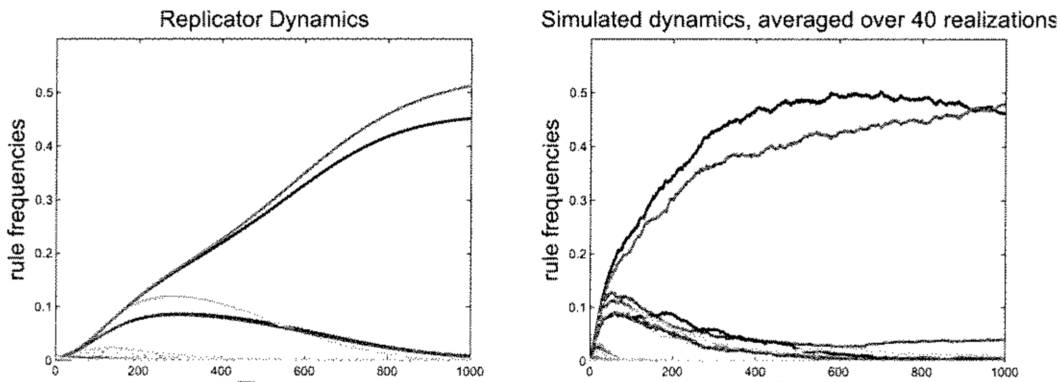
The effect of old strings on the result is limited as the high flow keeps the food pool fresh. As an upper limit the number of strings older than one is around 10% of the total food pool as seen in figure 8, calculated by assuming that all digestions succeed. In actual simulations the succeeded digestions are significantly fewer resulting in even less old strings. During the simulations to check stability of populations the number of strings older than one was measured. This number can be compared between populations that are predicted correctly and the ones that are predicted wrongly. Unfortunately the number of strings of age larger than one seem to be roughly the same for pairs with correct and incorrect predictions.

An even more likely cause for the discrepancy between the results of the replicator and the simulations is noise. It enters in many parts of the model. Noise in the measurements of interaction parameters is likely one cause for the difference in outcomes between simulation and replicator model. This noise is based in the randomness of food string generation (see section 3.3 to see how this is done). Since elementary CA are deterministic it the noise is not to big a problem for fresh strings. For multiple step digestions with the noise is present through all steps the number of possible outcomes increases. There is also the noise in the actual simulation to consider i.e. the underlying stochasticity of dividing, choosing strings etc. Now consider the situation where mutations are excluded, a rule that dies out won't be able to return to the population. For a population of rules with a stable fixed point there is still a probability that one rule might go extinct due to the random fluctuations. This is more likely in the case of populations where the stable points is close to one or zero and populations where the impact of cross-feeding is weaker. This error makes it possible for stable populations to collapse and is more likely to happen for longer simulations.

Returning to the populations that are predicted to be unstable but during simulations turn out to be stable there is one more effect that needs mentioning. Due to the underlying dynamics of Urdar a population of very poor rules that are unable to digest almost any string will have significantly longer relaxation time. The replacement of one individual with another only occurs after division and if very few successful digestions takes place the dynamics are very slow. Thus a population with parameters that predict that one rule takes over could in the actual simulation take very long time to do so. The number of falsely stable populations should decrease if the simulation runs for more iterations. This is the case for the populations that are stable as well as can be seen in table 2 when comparing the results after 4000 iterations with after 8000 iterations.

4.5.2 Full scale simulation

The replicator formulation is essentially a set of differential equations for the frequencies of all rules. Using the measured interaction data along with a all rules two player formulation replicator dynamics and solving this provides a way of estimating the full system dynamics at high flow rates. This comparison has been done and the result can be seen in figure 12. The simulated dynamics were averaged over several realizations. The overall results are similar the two best rules are the same in both simulated and replicator dynamics. There are bundles of less potent rules that are the same for the two different implementations. Some problems exist however. The first has been mention previously in section 4.4.5 and that is the difference in rate between replicator dynamics and simulated dynamics due to a difference in the fitness function. This is visible in the figure as difference between how fast the very weakest rules leave the population. A second difference lies in the measurements of the interaction terms and genetic drift. The two rules who are best both in simulated and replicator dynamics are 90 and 165. As their mappings are identical but with the bits flipped they are in fact equally good when contending against each other. As mentioned previously in section 4.4.4 this neutrality is not captured in the replicator dynamics if not the interaction terms are rounded to a set precision. In the simulated dynamics one of the two rules always wins due to drift but as seen on average they win the same number of times. In the current implementation of the replicator dynamics rule 165 always wins if solving the replicator equation for long times.



(a) Rule frequencies as derived when solving the replicator equation

(b) Rule frequencies as measured and averaged over 40 simulation runs

Figure 12: The two panels show the rule frequencies over time as given by solving the replicator equation and by measuring on the simulation. Each line represents a single rule. The exact identity of the rules are unimportant but note that the bundles or lines contain the same rules both in replicator and simulation. The difference in time scale is not entirely obvious but visible when considering the bundle of very weak rules that quickly die off. In the replicator dynamics these last until roughly $t = 400$ but in the simulation they are gone by $t = 100$. This is further discussed in section 4.4.5 but the time constant derived is not included in this solution of the replicator equation.

4.6 Optimality and the replicator model

Looking back to the replicator dynamics and considering instead the concept of optimality which was the main goal of this project there are a few possible paths to take. One obvious way to proceed would be to consider again the concept of global optimality. With basis in the replicator dynamics as defined in equation 11 it would be possible to analytically find the optimal population for the entire system. Given accurate measurements for all the interaction higher interaction terms this would be the ideal case. However due to time limitations this will not be possible withing the scope of this project. The previously derived two player two strategy formulation of the replicator equation can be used to do an initial evaluation of optimality in this context. First noting that the efficiency of a population is directly given by the average fitness

$$\overline{\phi} = x\phi_A + (1-x)\phi_B. \tag{31}$$

In the range [0 1] this function has two extreme points at the endpoints and a possible internal optimum. How well the final population coincides with the optimum population can be done by comparing the distance between the two populations. If this distance is small it indicates that the system does approach the optimum. Another way would be to calculate the ratio between efficiency of the final population and the efficiency at the optimum. If this ratio is near one it would also be an indication that high efficiency is a result of evolution.

As most pairs possible fall under the dominance category with one rule being clearly stronger the population consisting of 100% of that rule is also likely to be the optimum albeit trivially. To get results that say more about a cross-feeding population and how interactions effect optimality the investigation will focus on pairs of rules predicted to coexist. The approach is as follows. Calculate the mean fitness at the end points and in the possible internal optimum. Take the largest of these and divide it with the average fitness of the predicted internal fixed point. Average this value over all the pairs predicted to have stable internal fixed points. Of the 1187 unique pairs possible the average ratio is 0.95 with a standard deviation of 0.064. The average ratio between the least effective population and the most effective population were among the tested pairs 0.58 standard deviation of 0.34. This would indicate a rather strong link between the final population and the optimal population.

5 Discussion

The starting point of this project was to analyze the concept of optimality in evolution by studying the artificial life system Urdar. Starting from the question whether selection on the individual can lead to optimization on the population level. This question had no simple answer and a number of approaches were taken to evaluate it. First global optimization was applied to find the best population. This investigation did not yield clear results, good populations were found through the use of an evolutionary algorithm but no clearly best population appeared. What can be said at this point is that the populations that appear in the simulation do have high efficiency but not that they evolve towards a clearly best population.

In order to get more easily interpretable results the investigation turned to a reduced system. Even when looking at only a few rules at a time the results are inconclusive. There is a link between a population having high efficiency and being stable but the final population does not coincide with the best population. Instead through frequency dependent selection pushes the population away from the optimum. This reduced dynamics initiated a probing towards the depths of Urdar and the investigation of interaction strengths. By themselves they did not give to much information but they led to what has to be considered the main result of this project. Even thou only a byproduct of the original problem statement the replicator formulation of the Urdar dynamics provide a new way of characterizing the system.

The replicator formulation provides a way to analytically treat the dynamics. To formulate conditions for cooperation and stability. With accurate measurements of interactions between rules it also provides a way to predict behavior and generate the full dynamics. It also provides an intriguing game theoretic interpretation with a system of mixed games. Rules are partaking in one-, two-, three- etc. games continuously through the joint digestions of strings. With the exact mixture of games being controlled by the flow rate and the rules themselves through their ability to digest strings.

Through the replicator equation it is also possible to go back to the first question of optimality. However the later parts of the project were mainly devoted to testing the viability of the replicator equation by comparing it's predictions to the actual simulation. Results from the comparison of fixed points between pairs of rules were ambiguous in that they captured some stable fixed points but that many more appeared in the simulations. Still the thought is that through increased accuracy in measurements and by including more terms the results will be even better. The predictions and the brief check on optimality can be seen as hints on what will be possibly if the work is continued.

5.1 Future Work

There are many parts of this project that could be extended upon. Such as redoing the global optimization with another method or examining more triplets of rules. However the most promising development is the replicator formulation and it opens new avenues of investigation. The following is a list of a few of the extensions possible.

- Modify the current implementation of Urdar to make the analysis easier. One such item to change would be the probability function that controls division.
- Measure higher order interaction terms or possibly find another way of approximating them.

- Extend the replicator formalism and apply it to higher order interactions in Urdar.
- Investigate the replicator formulation as a standalone way of describing cross feeding systems. With a theoretical base for such systems it could then be applied to other systems similar to Urdar both real and artificial.

6 Conclusions

In Urdar there is a link between a population being efficient and that same population being stable. However the stable population is not by default the same as the optimal population and there is often some distance in between. In other systems the final population being the most productive population is often not the expected outcome. A famous example is the prisoners dilemma where the low scoring defection is the stable outcome.

The results presented here are not conclusive with regards to evolution leading to optimality even in this simplified setup. The next step would be to examine the proposed fitness equation. It can allows the system to be described in a more general form. An analytical expression also offers more transparency of causality compared to computer simulations. This could lead to better explanations of why or why not a population would evolve towards an optimal composition.

References

Altrock, P.M., Traulsen, A. (2009) Fixation times in evolutionary games under weak selection. *New Journal of Physics*, vol. 11, no. 1, DOI:10.1088/1367-2630/11/1/013012.

Belenguer, A., Duncan, S.H., Calder, A.G., Holtrop, G., Louis, P., Lobley, G.E., Flint, H.J. (2006) Two Routes of Metabolic Cross-Feeding between Bifidobacterium adolescentis and Butyrate-Producing Anaerobes from the Human Gut. *Applied and Environmental Microbiology*, vol. 72, no. 5, pp. 3593–3599, DOI:10.1128/AEM.72.5.3593-3599.2006.

Bull, J.J., Harcombe, W.R. (2009) Population dynamics contrain the cooperative evolution of cross-feeding. *PLoS ONE*, vol. 4, no. 1, DOI:10.1371/journal.pone.0004115.

Combach, A., Hogeweg, P. (2009) Evolution of resource cycling in ecosystems and individuals. *BMC Evolutionary Biology*, vol. 9, no. 122, DOI:10.1186/1471-2148-9-122.

Dejonghe, W., Berteloot, E., Goris, J., Boon, N., Crul, K., Maertens, S., Höfte, M., De Vos, P., Verstraete, W., Top, E.M. (2003) Synergistic degradation of linuron by a bacterial consortium and isolation of a single linuron-degrading Variovorax strain. *Applied and Environmental Microbiology*, vol. 69, no. 3, pp. 1532–1541, DOI:10.1128/AEM.69.3.1532-1541.2003.

Dittrich, P., Ziegler, J., Banzhaf, W. (2001) Artificial chemistries–a review. *Artificial Life*, vol. 7, no. 3, pp. 225–275.

Dupré, J. (1987) The latest on the best: essays on evolution and optimality. The MIT Press, Edited by John Dupré.

Estrela, S., Gudelj, I. (2010) Evolution of cooperative cross-feeding could be less challenging than originally thought. *PLoS ONE*, vol. 5, no. 11, DOI:10.1371/journal.pone.0014121.

G.A., Parker, Maynard Smith, J. (1990) Optimality theory in evolutionary biology. *Annual Review of Ecology and Systematics*, vol. 348, pp. 27–33, DOI:10.1038/348027a0.

Gardner, M. (1970) Mathematical games: the fantastic combinations of john conway's new solitare game "life". *Scientific American*, vol. 223, pp. 120–123.

Gerlee, P., Lundh, T. (2010a) Productivity and Diversity in a cross-feeding population of artificial organisms. *Evolution*, vol. 64, no. 9, pp. 2716–2730, DOI:10.1111/j.1558-5646.2010.01020.x.

Gerlee, P., Lundh, T. (2010b) Rock-scissor-paper dynamics in a digital ecology. *Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pp. 285–295.

Gokhale, C.S., Traulsen, A. (2010) Evolutionary games in the multiverse. *PNAS*, vol. 107, no. 12, pp. 5500–5504, DOI:10.1073/pnas.0912214107.

Hallam, S.J., Putnam, N., Preston, C.M., Detter, J.C., Rokhsar, D., Richardson, P.M., DeLong, E.F. (2004) Reverse methanogenesis: Testing the hypothesis with environmental genomics. vol. 305, no. 5689, pp. 1457–1462, DOI:10.1126/science.1100025.

Hamilton, W.D. (1964) The genetical evolution of social behaviour. I. *Journal of Theoretical Biology*, vol. 7, no. 1, pp. 1–16, DOI:10.1016/0022-5193(64)90038-4.

Katsuyama, C., Nakaoka, S., Takeuchi, Y., Tago, K., Hayatsu, M., Kato, K. (2009) Complementary cooperation between two syntrophic bacteria in pesticide degradation. *Journal of Theoretical Biology*, vol. 256, no. 4, pp. 644–654, DOI:doi:10.1016/j.jtbi.2008.10.024.

Kircher, J.W. (2002) The gaia hypothesis: fact, theory, and wishful thinking. *Climatic Ghange*, vol. 52, no. 4, pp. 391–408, DOI:10.1023/A:1014237331082.

Lovelock, J.E. (1972) Gaia as seen through the atmosphere. *Atmospheric Enviroment*, vol. 6, pp. 579–580.

Maynard Smith, J. (1978) Optimization theory in evolution. *Annual Review of Ecology and Systematics*, vol. 9, pp. 31–56, DOI:10.1146/annurev.es.09.110178.000335.

Maynard Smith, J. (1982) Evolution and the theory of games. Cambride, UK: Cambridge University Press.

Nash, J. (1950) Equilibrium points in n–person games. *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48–49.

Nowak, M.A. (2006) Five Rules for the Evolution of Cooperation. *Science*, vol. 314, no. 5805, pp. 1560–1563, DOI:10.1126/science.1133755.

Pfeiffer, T., Bonhoeffer, S. (2004) Evolution of cross-feeding in microbial populations. *Am Nat.*, vol. 163, no. 6, pp. E162–35.

Philips, J.D. (2008) Goal functions in ecosystem and biosphere evolution. *Progress in Physical Geography*, vol. 32, no. 1, pp. 51–64, DOI:10.1177/0309133308089497.

Rawlings, D.R., Johnson, D.B. (2007) The microbiology of biomining: development and optimization of mineral-oxidizing microbial consortia. *Microbiology*, vol. 153, no. 2, pp. 315–324, DOI:10.1099/mic.0.2006/001206-0.

Schelling, T.C. (1980) The strategy of conflict. Harvard University Press.

Shannon, C.E. (1948) A mathematical theory of communication. *The Bell System Technical Journal*, vol. 27.

Stams, Alfons J. M. (1994) Metabolic interactions between anaerobic bacteria in methanogenic environments. *Antonie van Leeuwenhoek*, vol. 66, no. 1, pp. 271–294, DOI:10.1007/BF00871644.

Ulam, S. (1962) On some mathematical problems connected with patterns of growth of figures. *Essays on Cellular Automata*.

von Neumann, J. (1928) Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, vol. 100, no. 1, pp. 295–320.

von Neumann, J., Burks, A.W. (1966) Theory of self–reproducing automata.

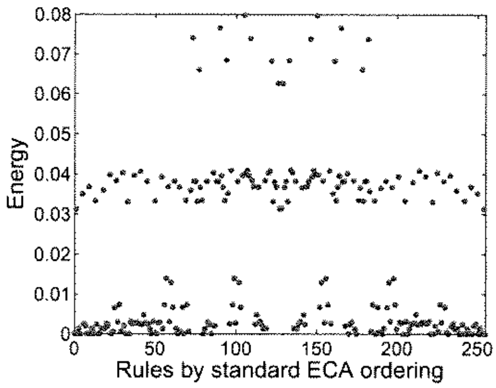
Watson, A.J., Lovelock, J.E. (1983) Biological homeostasis of the global environment: the parable of Daisyworld. *Tellus B*, vol. 35B, no. 4, pp. 284–289, DOI:10.1111/j.1600-0889.1983.tb00031.x.

Weissten, E. (2011) "linear stability.". From MathWorld–A Wolfram Web Resource., accessed: 2011-06-06 `http://mathworld.wolfram.com/LinearStability.html`.

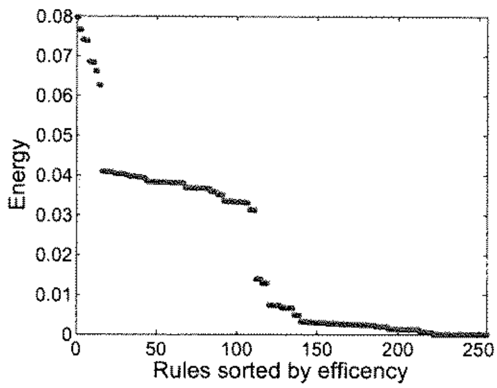
Whade, M. (2008) Biologically inspired optimization methods. Southampton, UK: WIT Press.

Wolfram, S. (1984) Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, vol. 10, no. 1-2, pp. 1–35, DOI:doi:10.1016/0167-2789(84)90245-8.

the final rule can extract. Here the concept of decreasing entropy becomes even more important. In order to properly separate the interaction coefficients, situations where the intention is to measure E_{ijk} but j or i fails to digest the string that realization is set to zero even though k might extract energy in the end. The intention is to let w_{ijk} describe exactly the average energy when rule k digests a string previously digested by i and j. When i or j fails to digest the actual energy becomes E_{jk} or E_{ik} which should not be included in the mean. The trials are still included in the mean for w_{ijk} but as zeros, since it should reflect the interaction strength between the rules. If rule k is able to extract high energy from strings previously digested by i and j but only manages this once in ten trials the interaction is still fairly weak.



(a) First interaction term w_i for all rules



(b) First interaction term w_i for all rules

Figure 5: Both the figures show the same data but ordered in different ways to give a feeling for the distribution of strengths. From a it is visible that pairs of mirrored rules are equally good at digesting strings. The best pair visible in a is 105 and 150. Also worth mentioning is the second best pair which is 90 and 165. Make a mental note that the top energy extracted is about 0.08 when digesting fresh strings.

Making the actual measurements takes considerable time. Single step interactions, w_i take less than a minute to complete for all rules. But for each additional interaction step we want to calculate the number of combinations increase with a factor of 256. In figure 5 the first order interaction terms can be seen for all rules. The division into three visible regions at ≈ 0.8 , ≈ 0.4 and ≈ 0 is likely caused by the chosen structure for fresh food where 50% are mostly zeros and the rest mostly ones. The three regimes would then be rules that can digest almost all strings, rules that can digest either black or white and rules that cant digest either. Figure 6 is an attempt to visualize all the second order interaction terms (w_{ij}). The aim of the figure is not to give an instant view of what rules have the strongest interactions but rather to give a notion of how the interaction space looks. The darker the pixel the higher the interaction value. The figure is not symmetric when reflected top to bottom but is so from right to left in the same way as in figure 5a. It is also possible to see similarities in 5a and figure 6 in that rules that are weak in the first digestion are also weak in the second. The white stripes across both directions in figure 6 are caused by such weak rules. The final thing to note, which is not possible to read from the figure, is that largest second order terms are as high as 0.14 compared to 0.8 for first order terms.

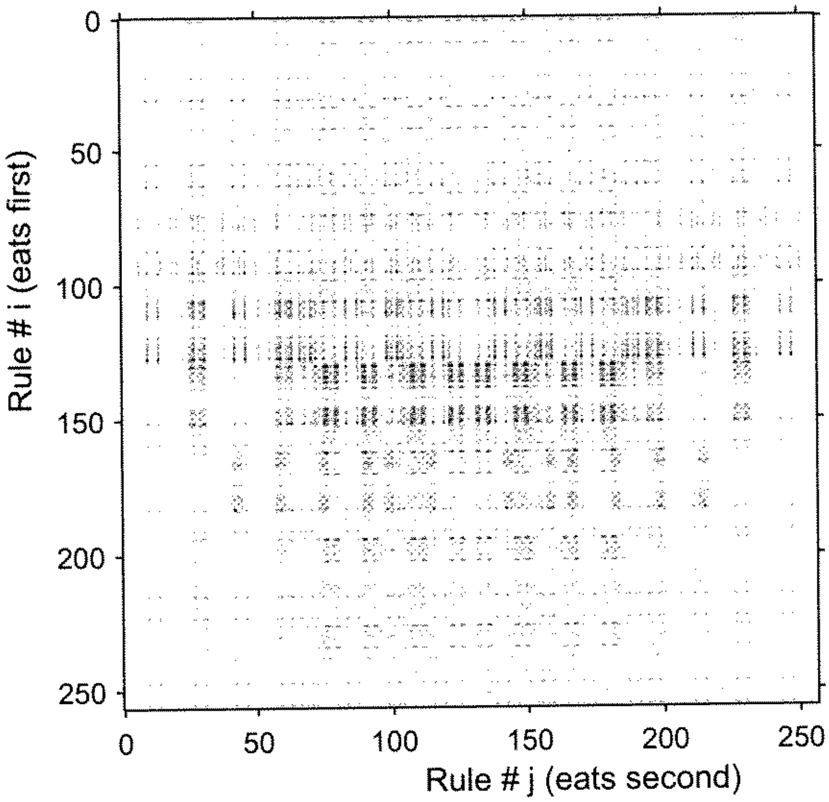


Figure 6: Second order interaction terms, w_{ij} measured for all rules. Dark pixels correspond to strong interactions i.e. high energy extracted by the second rule and white are very weak interactions. Shows similarities in structure to the first order interactions shown in figure 5a. Rules that are poor at digesting fresh strings are bad partners to interact with and that the strongest interactions are still found in the center region. Note, even thou it's not visible in the figure, that the highest second order interaction value is 0.14.

4.3.1 A first attempt at approximating interaction strengths

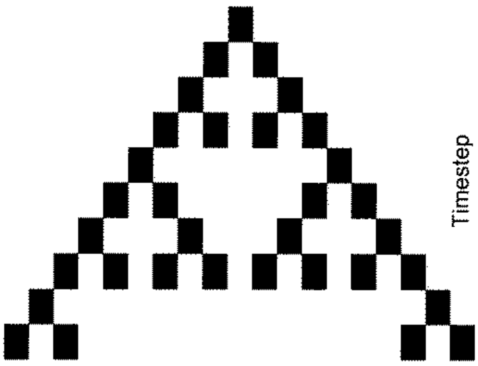
As stated previously the computational cost of calculating higher order interaction terms scales very badly. The number of possible combinations at for interactions of length n is 256^n. It is conceivable however that higher order terms might be approximated from lower order terms. Torbjörn Lundh proposed the following as a first attempt at doing this. In addition to the notation used above lets introduce the ratios between energy before and after digestion as $K_i = \frac{E_i}{E_0}$ and $K_{ij} = \frac{E_{ij}}{E_i}$. Now the assumption is that regardless of previous digestion histories relative energy uptake of a rule is roughly equal. That would mean that $K_{ij} \approx K_j$. Given some algebra this allows for a way to approximate higher order terms from lower order terms:

$$w_{ij} \approx \frac{E_0 - w_i}{E_0} w_j. \tag{9}$$

And for third order terms:

$$w_{ijk} \approx \frac{E_0 - w_i}{E_0} \frac{E_0 - w_j}{E_0} w_k. \tag{10}$$

The assumption has some merit but it turns out however that this approximation does not hold well. There are a few reasons for this. The first is that energy gain is not relative and can in fact be higher for the second digestion than for the first as show previously in figure 5 and 6. Figure 7 show the time evolution of ca-rule 90 and the corresponding entropy for each time step. The entropy differences between time steps two, one and zero do follow the assumption that extracted energy is relative to the energy left but after step two the assumption fails completely. This is an effect of how entropy is calculated and is non-obvious in terms of what patterns are intuitively expected to have high entropy. The key is that the entropy is calculated as the difference in block entropy between block size three and block size two as seen in equation 5. If a pattern exhibit an even mix of block types at level three and an even mix of patterns at block size two the approximated Shannon entropy is zero. Patterns of ...01010101... show exactly this behaviour with an even mix of 01 and 10 at size two and 010 and 101 at size three. This is why ca-rule 90 is not able to increase the entropy at time step three in figure 7.



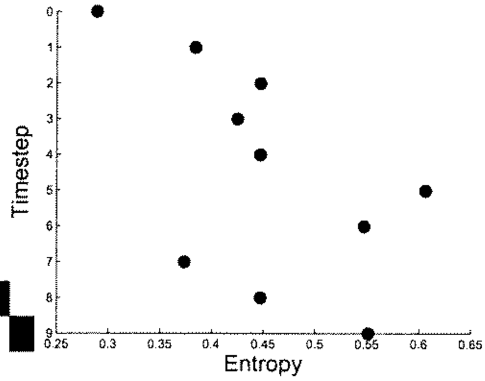


Figure 7: On the left are the first few time steps for rule 90 when starting from a single one history. On the right is the Shannon entropy approximated at level two for each time step. The key thing to note is that the entropy does not increase constantly over time instead there are plateaus. Since there is a restriction that a rule in Urdar is not allowed to decrease the entropy of a string rule 90 cannot fully digest this string alone. This decrease in entropy visible at step 3, 6 and 7 is caused by patterns that although containing an even mix of ones and zeros have roughly the same entropy at block size two and block size 3 thus having a low approximated Shannon entropy.

Nevertheless approximations of higher order terms might be fruitful and even necessary due to computational costs. Another approach would be to use second order interactions, w_{ij} etc, as a basis for the approximation. Even such a scheme might also fail however considering the example in figure 7. The rule performs well for two steps but is unable to digest the string again. This behavior would be hard to capture starting from w_{ij} terms. The approximation based on second order terms has not been tested however and is left to future work.

4.4 Urdar formulated as the replicator equation

With a framework to handle the how well rules perform together and alone many new paths of investigation open up. Philip Gerlee proposed the following formulation of a fitness function for ca rules in Urdar:

$$\phi_i = F(0)w_i + F(1)\sum_j x_j w_{ji} + F(2)\sum_k \sum_j x_j x_k w_{kji} + ... \tag{11}$$

$F(0)$ describes the fraction of fresh strings in the food pool, $F(1)$ describes the fraction of strings digested once(for more details see section 4.4.1). w_i, w_{ji} and w_{kji} are the interaction coefficients as described in section 4.3. x_j is the fraction of rule j in the population. Terms for longer interactions are included in a similar way.

The equation describes the fitness of rule i as a function of it's ability to digest fresh strings, it's ability to digest strings previously digested once weighed by the fraction of rules in the population and similar for longer interactions. Each term is also weighed by the fraction of strings of a certain age in the population. This distribution over string ages is of course strongly dependent on the dilution rate as described in section 4.4.1.

There are a few assumptions that might not be obvious that underlie this description of the fitness. One assumption that isn't clearly stated is that dynamics of the food pool must work on a shorter timescale than the population dynamics. What this means is that if the population shifts the strings in the food pool must be quickly replaced with fresh strings that are then digested to make the digestion histories of strings in the pool properly reflect the population structure. In the above formulation it would also appear that the string age distribution is constant regardless of population structure which is far from true when considering all populations. A population of weak rules(for example the ones with low score in fig ref) will have a much fresher food pool as they are often unable to digest the strings. In the subset of strong populations the assumption still holds however and since these are the ones of interest the assumption is reasonable.

The fitness function defined in 11 can now be used in a replicator equation to provide a way to generate the full population dynamics. The replicator equation defines the rate of change for a rule to be proportional the current fraction of that rule times it's fitness relative to the average fitness in the population.

$$\frac{dx_i}{dt} = x_i(\phi_i - \overline{\phi}). \tag{12}$$

The average fitness is defined as:

$$\overline{\phi} = \sum_j \phi_j x_j. \tag{13}$$

A rule that has higher fitness than the average fitness will increase it's fraction of the population while a rule with less than average fitness will decrease. Note the above formulation mutations between rules is not included.

4.4.1 Age distribution for food strings

In order to assign values to the $F(\cdot)$ weights in the previously described fitness function (ref first replicator) we need to know the age distribution for food strings. The age of a string corresponds to the number of times it has been digested. The dynamics of the food pool was described in section 3.3. As previously mentioned there are two competing processes that affect string age; strings being eaten and the flow of strings through the system. These processes can be treated analytically. First we note that in the current version of Urdar the number of strings in the food-pool is 5000 and the number of organisms is 1024. This gives a ratio of approximately $\frac{1}{5}$ that can be digested each time step. Assuming that all digestions succeed and that the distribution has reached a steady state the fraction of strings of a specific age can be described in the following way.

$$F(n) = \frac{1}{5}F(n-1) + F(n) - \frac{1}{5}F(n) - \gamma F(n). \tag{14}$$

The equation describes the fraction of strings of age n at time t as a function of string fractions at time t-1 but since steady state is assumed the time dependence is excluded. In order the right hand side terms describe: strings of age n-1 that get digested, strings of age n that stays, strings of age n that get digested and strings of age n that flows out of the system. The equation can be simplified into:

$$F(n) = \frac{F(n-1)}{1+5\gamma}. \tag{15}$$

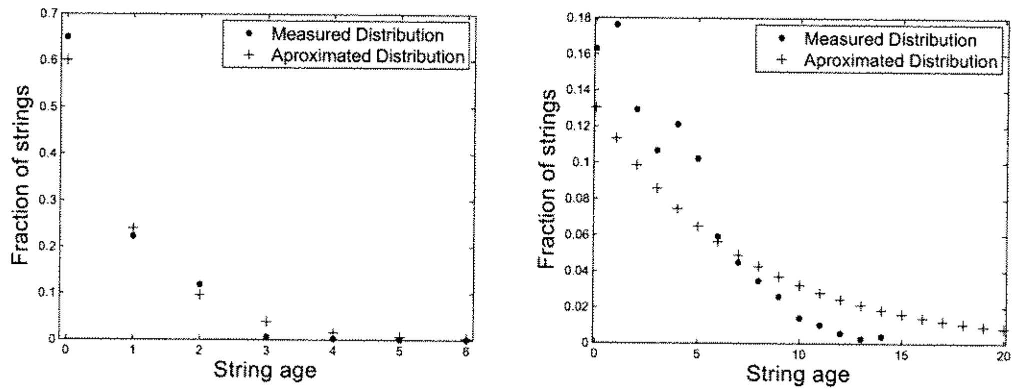
Now noting the special case of fresh strings, $n = 0$, derived from $F(0) = F(0) - \frac{1}{5}F(0) + \gamma(1 - F(0))$:

$$F(0) = \frac{5\gamma}{1+5\gamma}. \tag{16}$$

Going back 15 and inserting 16 it is possible to form the general expression:

$$F(n) = \frac{5\gamma}{(1+5\gamma)^{n+1}}. \tag{17}$$

In figure 8 a comparison of the distribution of ages in the food-pool between what



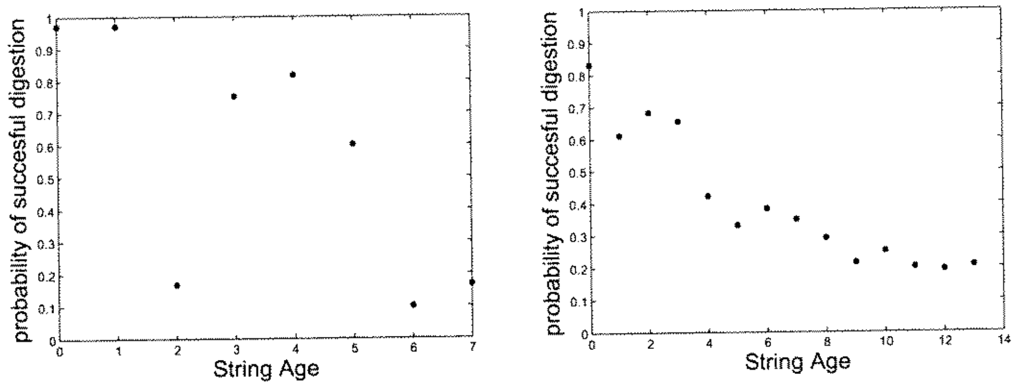
(a) Distribution of string ages for flow 0.3 (b) Distribution of string ages for flow 0.03

Figure 8: The measured and approximated distributions over string ages. Note the difference in scale for the two examples. For flow 0.3 in 8a there is a large amount of fresh strings and the approximation fits the measured data well. For flow 0.03 8b the approximated distribution does not fit the measurements as well and the strings are fresher than expected. The main reason for this is that the probability for digestion is not 100% and not uniform over string ages. On the contrary it drops rapidly as seen in figure 9. This lowers the rate at which strings are digested which in turns means that fewer strings achieve a high age before being flushed out.

is generated by the expression in 17 and what is measured in an actual simulation can be seen. The cause for the difference between the measured distribution and the approximated lies in the assumption that all digestions succeed.

4.4.2 Game theoretic interpretation

The above reformulation of the Urdar dynamics can be seen as a game theoretic formulation of the interactions. In Urdar the strategies correspond to rules and the number of players are the interaction length. Essentially we have a population of strategies or rules simultaneously partaking in games with different number of players. The digestion of a fresh string corresponds to a trivial one player game where a rule simply receives payoff based on itself. Digestions of strings previously digested once become two player games and three digestions become three player



(a) Probability for successful digestions at flow 0.3 (b) Probability for successful digestions at 0.03

Figure 9: The probability of successful digestions for strings of different ages. Neither of the two cases seen correspond well to the approximation that all digestions are successful. Still for the case of high flow and strings of age zero and one the approximation fit well and as a result the the calculated and measured data in figure 8a to be very similar.

games etc. This distribution over games of different number of players is determined by the flow rate as described in section 4.4.1. The tensor notation described in section 2.3.1 now becomes equivalent to the one used describe interaction terms with a slight adjustment to formalism. While the tensor notation used in (Gokhale and Traulsen, 2010), $\beta_{i_0,i_1,i_2,\ldots,i_{d-1}}$, put the focal strategy first the notation used for interaction terms, $w_{ijk\ldots l}$, lets the last index notate the strategy receiving the payoff.

4.4.3 Replicator equation for two rules

Consider a population consisting of only two rules A and B their fractions denoted by x and y. Since the total population sums to one and the part that isn't x is y it is possible to reformulate $y = 1 - x$. Given this it is possible to reformulate the replicator equation in 12 into:

$$\frac{dx}{dt} = x(1-x)(\phi_A - \phi_B) = P(x). \tag{18}$$

Leaving a single differential equation to determine the system. Now arguing that we want to examine a system of two rules at high dilution rates it is possible to throw away interactions of length three and higher. It is then possible to write the fitness functions for A and B as:

$$\phi_A = F(0)w_A + F(1)(w_{BA}(1-x) + w_{AA}x), \tag{19}$$

$$\phi_B = F(0)w_B + F(1)(w_{AB}x + w_{BB}(1-x)). \tag{20}$$

This setup allows for analytical treatment of the system. That is locating and characterizing fixed points and examining productivity which among other things allows for prediction on which rules can exhibit stable cooperation in the actual simulation. First noting that a fixed point is defined as point in the state space where the rate of change is zero $\frac{dx}{dt} = 0$. Looking at equation 18 it has two trivial fixed points at $x = 1$ and $x = 0$ corresponding to the entire population consisting of 100% A or 100% B. There is also a fixed point at x^* when:

$$\phi_A = \phi_B. \tag{21}$$

The position of this fixed point is important to predict the dynamics of any given rule pair. If this fixed point lies withing the range $x^* \in (0\ 1)$ and is stable it means that the two rules can coexist. But leaving stability for later lets first look at the position of this fixed point. By expanding equation 21 and solving it for x the following expression for the position of the fixed point is derived.

$$x^* = \frac{\frac{F(0)}{F(1)}(w_B - w_A) + w_{BB} - w_{BA}}{w_{AA} + w_{BB} - w_{AB} - w_{BA}}. \tag{22}$$

It is possible to devise conditions for when the fixed point lies within $x^* \in (01)$ but first consider the concept of stability for fixed points. Consider the situation where system resides at the fixed point. The system now receives a small perturbation and is pushed away from the fixed point. If the dynamics drives the system further away from the fixed point it is considered unstable. If instead it is drawn back to the fixed point it is stable. In more formal terms the stability is investigated by linear stability analysis which corresponds expanding the set of differential equations to the first order at the fixed point. This yields a stability matrix and the eigenvalues of this matrix determines the stability (Weissten, 2011). For this one dimensional case however this the stability matrix simply becomes a scalar value corresponding to the derivative of equation 18 and the sign of this value will determine the stability. If positive any perturbation will grow exponentially over time corresponding to an unstable fixed point and if negative perturbations will decrease exponentially corresponding to a stable fixed point. Calculating the time derivative of equation 18 yields:

$$P'(x) = (1-x)(\phi_A - \phi_B) - x(\phi_A - \phi_B) + x(1-x)(\phi'_A - \phi'_B). \tag{23}$$

Given that equation 21 holds this reduces to:

$$P'(x^*) = x(1-x)(\phi'_A - \phi'_B). \tag{24}$$

Noting that $x \in (01)$ makes sure that $x(1-x) > 0$ thus the only term influencing the sign of $P'(x)$ is the derivatives of the fitness functions.

$$\phi'_A - \phi'_B = F(1)(w_{AA} - w_{BA}) - F(1)(w_{AB} - w_{BB}).$$

Noting that $F(1)$ is a positive constant the expression above is negative if the following holds.

$$w_{AB} + w_{BA} < w_{AA} + w_{BB}. \tag{25}$$

This equation captures the stability criteria for the internal fixed point. It has an intuitive interpretation. If the two rules gains more from eating each others leftovers than they do from eating their own leftovers the internal fixed point will be stable. That is to say they are able to live together granted that the fixed point is in fact internal. Moving back to equation 22 it is possible to state criteria for when the fixed point is internal i.e. $0 < x^* < 1$.

$$0 < \frac{\frac{F(0)}{F(1)}(w_B - w_A) + w_{BB} - w_{BA}}{w_{AA} + w_{BB} - w_{AB} - w_{BA}} < 1.$$

If this is to be simplified the sign of the denominator $w_{AA} + w_{BB} - w_{AB} - w_{BA}$ needs to be determined. Conveniently this corresponds to the stability criteria for the fixed point. When looking for the stable internal fixed points the denominator is negative and for the unstable fixed points the denominator is positive. This allows the following separation of the two cases. For stable internal fixed points the criteria becomes.

$$F(1)(w_{AA} - w_{AB}) < F(0)(w_B - w_A) < F(1)(w_{BA} - w_{BB}). \tag{26}$$

And for the internal but unstable fixed points.

$$F(1)(w_{AA} - w_{AB}) > F(0)(w_B - w_A) > F(1)(w_{BA} - w_{BB}). \tag{27}$$

These criteria also contain some intuitive interpretations regarding the existence of the internal fixed point. The left hand side describes the relationship between A consuming A's leftovers and B consuming A's leftovers. If the w_{AB} term is much larger than the w_{AA} term the lhs will be small. This would increase the range of possible values for the center expression that give a stable fixed point but decrease them for the unstable fixed point. That would agree with the notion that large cross-feeding terms promote coexistence. The same argument holds for the right hand side. A large w_{BA} term compared to the w_{BB} term promotes stability.

There is still two fixed points that have not been characterized for this simplified system namely $x = 0$ and $x = 1$. The stability of these fixed points is also determined by the sign of $P'(x)$. For $x = 0$ equation 23 reduces to.

$$P'(0) = (\phi_A - \phi_B).$$

And for $x = 1$ it reduces to.

$$P'(1) = -(\phi_A - \phi_B).$$

Inserting $x = 0$ and $x = 1$ into equations 19 and 20 which describes the fitness functions reduces the above to.

$$P'(0) = F(0)(w_A - w_B) - F(1)(w_{BB} - w_{BA}).$$

And for $x = 1$ it reduces to.

$$P'(1) = F(0)(w_B - w_A) - F(1)(w_{AA} - w_{AB}).$$

Constraining the above expressions to be negative we arrive at criteria for when $x = 0$ and $x = 1$ are stable fixed points. For $x = 0$:

$$F(0)(w_A - w_B) < F(1)(w_{BB} - w_{BA}). \tag{28}$$

For $x = 1$:

$$F(0)(w_B - w_A) < F(1)(w_{AA} - w_{AB}). \tag{29}$$

Again the resultant criteria are intuitively understandable. If for example both w_B and w_{BB} are large rule B is both good at eating fresh strings and it's leftovers. This would also indicate according to equation 28 that $x = 0$, corresponding to a population of 100% B, is a stable fixed point which is the expected result. The same reasonings holds for the fixed point at $x = 1$ with focus on rule A.

Note that the criteria for endpoint stability are not mutually exclusive. Both endpoints can be stable or unstable at the same time. This corresponds to the situation where there is an internal fixed point and it is unstable or stable. It turns out that the criterion for unstable internal fixed point and criterion that both endpoints are stable are exactly equivalent. By rewriting equation 28 into

$$F(0)(w_B - w_A) > F(1)(w_{BA} - w_{BB}).$$

and combining this with equation 29 the following is attained.

$$F(1)(w_{AA} - w_{AB}) > F(0)(w_B - w_A) > F(1)(w_{BA} - w_{BB}).$$

This is exactly the condition for existence of an unstable internal fixed point. The same holds when looking at an internal stable fixed point and both endpoints unstable. This result is no surprise however as in a continuous system it is impossible to have two stable or two unstable fixed points next to each other. Consider the case with two stable fixed points next to each other. Standing in between the two you get pulled into one or the other depending on the distance. However as the force driving the system is continuous there must exist a point in between the two fixed points where their pull is exactly equal. If standing at this point you would remain there indefinitely but take a small step to either side and one of the stable points would pull you in. Which is exactly the definition of an unstable fixed point. Going back to the criteria for the fixed points in the two rule replicator model it is expected that the criterion for an internal stable fixed point is equivalent to the criterion that both endpoints are unstable.

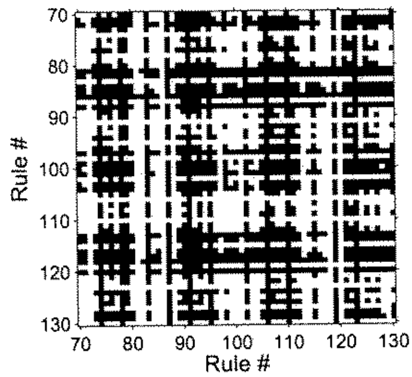
Using the measured data for first and second order interaction shown in figures terms it is possible to apply the derived criteria to predict pairs of rules exhibiting the different kinds of fixed points.

4.4.4 Characterization of two rule dynamics

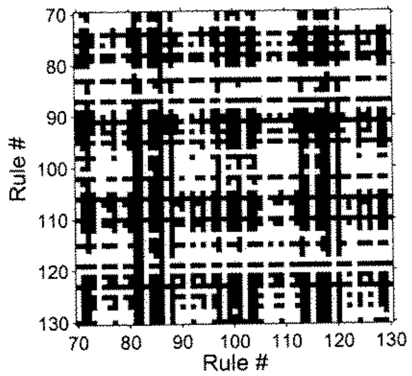
Based on the stability and localization of fixed points the dynamics for two rules can belong to one of four different categories exemplified in (Altrock and Traulsen, 2009). Dominance means that one rule is always the winner regardless of starting position corresponding one stable and one unstable endpoint. Coexistence allows the two rules to live together and is characterized by a stable internal fixed point. Coordination games takes place when there is an unstable internal fixed point and the outcome depends on the starting condition. There is also a situation not mentioned in (Altrock and Traulsen, 2009) and that is the neutral situation with two rules or strategies being equally good. The situation of neutrality is not natural in the circumstances. It would in theory occur between pairs rules whose mappings are inverted for example 90 and 165. These are equally good at extracting energy from strings as discussed in section 2.1. Two things complicate this however. As the Urdar simulation is a stochastic process there is always drift in the population and two equally good rules can still diverge. The second issue is that the interaction terms used in the replicator formulation are measured averages and unless they are rounded to a fixed precision it wont be able to capture true neutrality. Table 1 show the number pairs for each of the four types as predicted by the two rule two player formulation of the replicator dynamics.

4.4.5 Rate difference between replicator representation and simulated dynamics

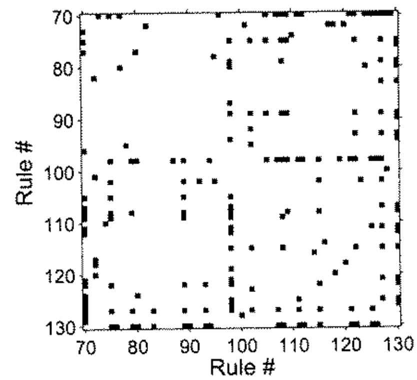
In the replicator dynamics described above the ability to extract energy is taken directly as the fitness. In the simulation however the energy is used as an input to a probability function as shown in equation 6 in section 3.2. The fitness in the simulation instead of being simply energy extracted follows this equation. With $\beta = 10$ this translates to a rate difference when comparing simulation with replicator



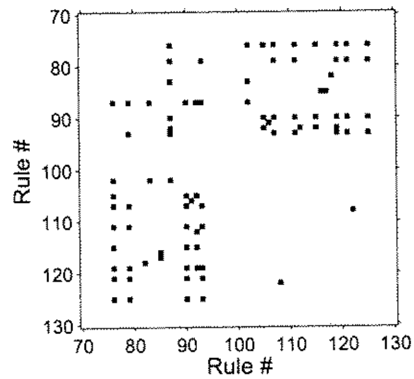
(a) Rule pairs predicted to have x = 0 as a stable fixed point



(b) Rule pairs predicted to have x = 1 as a stable fixed point



(c) Rule pairs predicted to have a stable internal fixed point



(d) Rule pairs predicted to have an unstable internal fixed point

Figure 10: Four plots showing the predicted distribution of fixed points for a subset of all pairs. Black pixels indicate pairs with a stable fixed point except in panel d. Interesting to note is the overlap between distributions. For example a pair with stable fixed points in both panel a and b will also exhibit an unstable fixed point in panel d. Pairs on the diagonal corresponding to picking the same rule twice are trivial.

dynamics. One way to account for this is to approximate equation 6 for small ΔE by Taylor expansion about zero.

$$\frac{1-e^{-\Delta E\beta}}{1-e^{-\beta}} \approx \frac{1-(1-\beta\Delta E+...)}{1-e^{-\beta}}.$$

Including only the first order term and noting that e^{-10} is a very small number equation 6 reduces to.

$$P(\Delta E) \approx \beta\Delta E. \tag{30}$$

However since $\beta = 10$ and ΔE often is as large as 0.1 this is a rather poor approximation. However as a first attempt at accounting for the rate differences that will exemplified in the following section this is sufficient.

4.5 Test of the replicator dynamics

The replicator dynamics has been presented as a way to reduce the Urdar system to a more manageable analytical formulation. It is possible to examine requirements for

Type	Number of pairs
Dominance	30304
Coexistence	1187
Coordination	1139

Table 1: The number of pairs exhibiting the four interaction types. The total number of pairs is $\frac{256^2-256}{2} = 32640$ not counting duplicates and rules paired with themselves. 128 of the pairs above should in reality fall under the neutrality category but they can not be captured by the current measurements of interaction terms. Summing the classes together gives a number 10 short of the expected total. This is caused by a few number of rules who have $w_i = 0$ which cause the replicator notation to collapse to zero. Pairs of of these rules can be considered to belong to the neutral category.

coexistence and predict behavior. Given accurate measurements for interaction terms it is possible to generate the full system dynamics by solving the replicator equation. As only the first two interaction terms have been measured it is not possible to test higher order dynamics. Still the Urdar system at high flow rates is driven mainly by the digestion of fresh and one digestion old strings. This makes it possible to use the replicator formulation to as an approximation to the Urdar system at high flow rates. The following subsections deal with testing on how well the replicator equation capture the simulated dynamics.

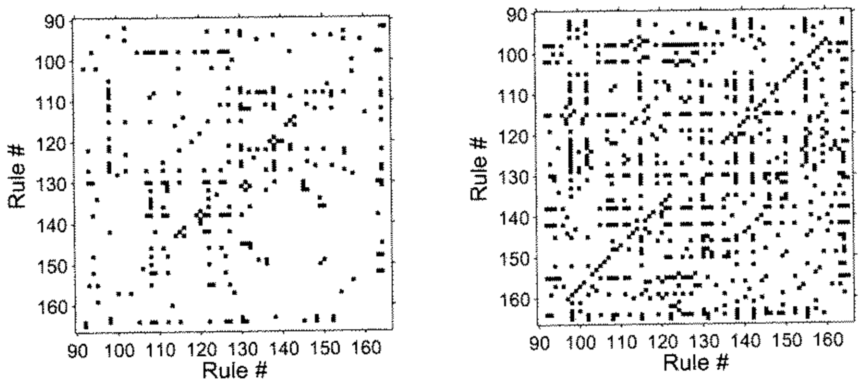
4.5.1 Predicted fixed points for two rules

The replicator formulation of Urdar with two players and two rules show in section 4.4.3 was used to predict rules with stable internal fixed points. It is possible to test these predictions through simulation of the pairs. Letting the two rules live together for many iterations and then checking whether both still are present in the population. This was done for a subset of all possible pairs namely rules in the range 90-165. The result of this testing can be seen in table 2. A visual comparison between predicted and tested fixed points can be seen in figure 11.

Type	Predicted	After 4000 iterations	After 8000 iterations
Stable fixed point	141	121	115
Stable, tested as unstable		20	26
Unstable fixed point	2894	2769	2894
Unstable, tested as stable		270	239

Table 2: Summary of the result when testing fixed point stability. The tests were conducted by simulating the populations for the given time period and checking if both rules survived. The total number of fixed points found to be stable is 391 and 354 for the two time spans is much higher than the predicted number of fixed points. Increasing the number of iterations decreases the number of stable fixed points.

There are a number of possible causes for a population with a predicted stable fixed point to collapse or a population predicted to be unstable to survive. The first cause is the higher order terms that have been excluded in this first representation. Two rules who do not benefit enough from cross-feeding strings that have been digested once can plausibly find more opportunities for cross-feeding on older strings. In a similar fashion populations on the verge of coexistence could collapse due to one rule being able to consistently re-digest it's own waste.



(a) Rule pairs predicted to have a stable internal fixed point

(b) Rule pairs observed to have a stable internal fixed point

Figure 11: Rules exhibiting stable internal fixed points as predicted and tested. Black pixels correspond to rules with stable fixed points. Main thing to note is the great increase in fixed points found when testing.

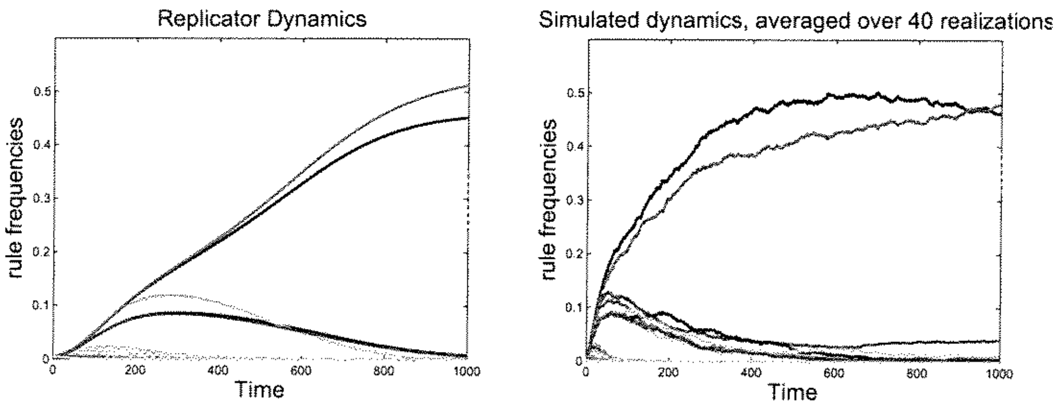
The effect of old strings on the result is limited as the high flow keeps the food pool fresh. As an upper limit the number of strings older than one is around 10% of the total food pool as seen in figure 8, calculated by assuming that all digestions succeed. In actual simulations the succeeded digestions are significantly fewer resulting in even less old strings. During the simulations to check stability of populations the number of strings older than one was measured. This number can be compared between populations that are predicted correctly and the ones that are predicted wrongly. Unfortunately the number of strings of age larger than one seem to be roughly the same for pairs with correct and incorrect predictions.

An even more likely cause for the discrepancy between the results of the replicator and the simulations is noise. It enters in many parts of the model. Noise in the measurements of interaction parameters is likely one cause for the difference in outcomes between simulation and replicator model. This noise is based in the randomness of food string generation (see section 3.3 to see how this is done). Since elementary CA are deterministic it the noise is not to big a problem for fresh strings. For multiple step digestions with the noise is present through all steps the number of possible outcomes increases. There is also the noise in the actual simulation to consider i.e. the underlying stochasticity of dividing, choosing strings etc. Now consider the situation where mutations are excluded, a rule that dies out won't be able to return to the population. For a population of rules with a stable fixed point there is still a probability that one rule might go extinct due to the random fluctuations. This is more likely in the case of populations where the stable points is close to one or zero and populations where the impact of cross-feeding is weaker. This error makes it possible for stable populations to collapse and is more likely to happen for longer simulations.

Returning to the populations that are predicted to be unstable but during simulations turn out to be stable there is one more effect that needs mentioning. Due to the underlying dynamics of Urdar a population of very poor rules that are unable to digest almost any string will have significantly longer relaxation time. The replacement of one individual with another only occurs after division and if very few successful digestions takes place the dynamics are very slow. Thus a population with parameters that predict that one rule takes over could in the actual simulation take very long time to do so. The number of falsely stable populations should decrease if the simulation runs for more iterations. This is the case for the populations that are stable as well as can be seen in table 2 when comparing the results after 4000 iterations with after 8000 iterations.

4.5.2 Full scale simulation

The replicator formulation is essentially a set of differential equations for the frequencies of all rules. Using the measured interaction data along with a all rules two player formulation replicator dynamics and solving this provides a way of estimating the full system dynamics at high flow rates. This comparison has been done and the result can be seen in figure 12. The simulated dynamics were averaged over several realizations. The overall results are similar the two best rules are the same in both simulated and replicator dynamics. There are bundles of less potent rules that are the same for the two different implementations. Some problems exist however. The first has been mention previously in section 4.4.5 and that is the difference in rate between replicator dynamics and simulated dynamics due to a difference in the fitness function. This is visible in the figure as difference between how fast the very weakest rules leave the population. A second difference lies in the measurements of the interaction terms and genetic drift. The two rules who are best both in simulated and replicator dynamics are 90 and 165. As their mappings are identical but with the bits flipped they are in fact equally good when contending against each other. As mentioned previously in section 4.4.4 this neutrality is not captured in the replicator dynamics if not the interaction terms are rounded to a set precision. In the simulated dynamics one of the two rules always wins due to drift but as seen on average they win the same number of times. In the current implementation of the replicator dynamics rule 165 always wins if solving the replicator equation for long times.



(a) Rule frequencies as derived when solving the replicator equation

(b) Rule frequencies as measured and averaged over 40 simulation runs

Figure 12: The two panels show the rule frequencies over time as given by solving the replicator equation and by measuring on the simulation. Each line represents a single rule. The exact identity of the rules are unimportant but note that the bundles or lines contain the same rules both in replicator and simulation. The difference in time scale is not entirely obvious but visible when considering the bundle of very weak rules that quickly die off. In the replicator dynamics these last until roughly $t = 400$ but in the simulation they are gone by $t = 100$. This is further discussed in section 4.4.5 but the time constant derived is not included in this solution of the replicator equation.

4.6 Optimality and the replicator model

Looking back to the replicator dynamics and considering instead the concept of optimality which was the main goal of this project there are a few possible paths to take. One obvious way to proceed would be to consider again the concept of global optimality. With basis in the replicator dynamics as defined in equation 11 it would be possible to analytically find the optimal population for the entire system. Given accurate measurements for all the interaction higher interaction terms this would be the ideal case. However due to time limitations this will not be possible withing the scope of this project. The previously derived two player two strategy formulation of the replicator equation can be used to do an initial evaluation of optimality in this context. First noting that the efficiency of a population is directly given by the average fitness

$$\overline{\phi} = x\phi_A + (1-x)\phi_B. \tag{31}$$

In the range [0 1] this function has two extreme points at the endpoints and a possible internal optimum. How well the final population coincides with the optimum population can be done by comparing the distance between the two populations. If this distance is small it indicates that the system does approach the optimum. Another way would be to calculate the ratio between efficiency of the final population and the efficiency at the optimum. If this ratio is near one it would also be an indication that high efficiency is a result of evolution.

As most pairs possible fall under the dominance category with one rule being clearly stronger the population consisting of 100% of that rule is also likely to be the optimum albeit trivially. To get results that say more about a cross-feeding population and how interactions effect optimality the investigation will focus on pairs of rules predicted to coexist. The approach is as follows. Calculate the mean fitness at the end points and in the possible internal optimum. Take the largest of these and divide it with the average fitness of the predicted internal fixed point. Average this value over all the pairs predicted to have stable internal fixed points. Of the 1187 unique pairs possible the average ratio is 0.95 with a standard deviation of 0.064. The average ratio between the least effective population and the most effective population were among the tested pairs 0.58 standard deviation of 0.34. This would indicate a rather strong link between the final population and the optimal population.

5 Discussion

The starting point of this project was to analyze the concept of optimality in evolution by studying the artificial life system Urdar. Starting from the question whether selection on the individual can lead to optimization on the population level. This question had no simple answer and a number of approaches were taken to evaluate it. First global optimization was applied to find the best population. This investigation did not yield clear results, good populations were found through the use of an evolutionary algorithm but no clearly best population appeared. What can be said at this point is that the populations that appear in the simulation do have high efficiency but not that they evolve towards a clearly best population.

In order to get more easily interpretable results the investigation turned to a reduced system. Even when looking at only a few rules at a time the results are inconclusive. There is a link between a population having high efficiency and being stable but the final population does not coincide with the best population. Instead through frequency dependent selection pushes the population away from the optimum. This reduced dynamics initiated a probing towards the depths of Urdar and the investigation of interaction strengths. By themselves they did not give to much information but they led to what has to be considered the main result of this project. Even thou only a byproduct of the original problem statement the replicator formulation of the Urdar dynamics provide a new way of characterizing the system.

The replicator formulation provides a way to analytically treat the dynamics. To formulate conditions for cooperation and stability. With accurate measurements of interactions between rules it also provides a way to predict behavior and generate the full dynamics. It also provides an intriguing game theoretic interpretation with a system of mixed games. Rules are partaking in one-, two-, three- etc. games continuously through the joint digestions of strings. With the exact mixture of games being controlled by the flow rate and the rules themselves through their ability to digest strings.

Through the replicator equation it is also possible to go back to the first question of optimality. However the later parts of the project were mainly devoted to testing the viability of the replicator equation by comparing it's predictions to the actual simulation. Results from the comparison of fixed points between pairs of rules were ambiguous in that they captured some stable fixed points but that many more appeared in the simulations. Still the thought is that through increased accuracy in measurements and by including more terms the results will be even better. The predictions and the brief check on optimality can be seen as hints on what will be possibly if the work is continued.

5.1 Future Work

There are many parts of this project that could be extended upon. Such as redoing the global optimization with another method or examining more triplets of rules. However the most promising development is the replicator formulation and it opens new avenues of investigation. The following is a list of a few of the extensions possible.

- Modify the current implementation of Urdar to make the analysis easier. One such item to change would be the probability function that controls division.
- Measure higher order interaction terms or possibly find another way of approximating them.

- Extend the replicator formalism and apply it to higher order interactions in Urdar.
- Investigate the replicator formulation as a standalone way of describing cross feeding systems. With a theoretical base for such systems it could then be applied to other systems similar to Urdar both real and artificial.

6 Conclusions

In Urdar there is a link between a population being efficient and that same population being stable. However the stable population is not by default the same as the optimal population and there is often some distance in between. In other systems the final population being the most productive population is often not the expected outcome. A famous example is the prisoners dilemma where the low scoring defection is the stable outcome.

The results presented here are not conclusive with regards to evolution leading to optimality even in this simplified setup. The next step would be to examine the proposed fitness equation. It can allows the system to be described in a more general form. An analytical expression also offers more transparency of causality compared to computer simulations. This could lead to better explanations of why or why not a population would evolve towards an optimal composition.

References

Altrock, P.M., Traulsen, A. (2009) Fixation times in evolutionary games under weak selection. *New Journal of Physics*, vol. 11, no. 1, DOI:10.1088/1367-2630/11/1/013012.

Belenguer, A., Duncan, S.H., Calder, A.G., Holtrop, G., Louis, P., Lobley, G.E., Flint, H.J. (2006) Two Routes of Metabolic Cross-Feeding between Bifidobacterium adolescentis and Butyrate-Producing Anaerobes from the Human Gut. *Applied and Environmental Microbiology*, vol. 72, no. 5, pp. 3593–3599, DOI:10.1128/AEM.72.5.3593-3599.2006.

Bull, J.J., Harcombe, W.R. (2009) Population dynamics contrain the cooperative evolution of cross-feeding. *PLoS ONE*, vol. 4, no. 1, DOI:10.1371/journal.pone.0004115.

Combach, A., Hogeweg, P. (2009) Evolution of resource cycling in ecosystems and individuals. *BMC Evolutionary Biology*, vol. 9, no. 122, DOI:10.1186/1471-2148-9-122.

Dejonghe, W., Berteloot, E., Goris, J., Boon, N., Crul, K., Maertens, S., Höfte, M., De Vos, P., Verstraete, W., Top, E.M. (2003) Synergistic degradation of linuron by a bacterial consortium and isolation of a single linuron-degrading Variovorax strain. *Applied and Environmental Microbiology*, vol. 69, no. 3, pp. 1532–1541, DOI:10.1128/AEM.69.3.1532-1541.2003.

Dittrich, P., Ziegler, J., Banzhaf, W. (2001) Artificial chemistries–a review. *Artificial Life*, vol. 7, no. 3, pp. 225–275.

Dupré, J. (1987) The latest on the best: essays on evolution and optimality. The MIT Press, Edited by John Dupré.

Estrela, S., Gudelj, I. (2010) Evolution of cooperative cross-feeding could be less challenging than originally thought. *PLoS ONE*, vol. 5, no. 11, DOI:10.1371/journal.pone.0014121.

G.A., Parker, Maynard Smith, J. (1990) Optimality theory in evolutionary biology. *Annual Review of Ecology and Systematics*, vol. 348, pp. 27–33, DOI:10.1038/348027a0.

Gardner, M. (1970) Mathematical games: the fantastic combinations of john conway's new solitare game "life". *Scientific American*, vol. 223, pp. 120–123.

Gerlee, P., Lundh, T. (2010a) Productivity and Diversity in a cross-feeding population of artificial organisms. *Evolution*, vol. 64, no. 9, pp. 2716–2730, DOI:10.1111/j.1558-5646.2010.01020.x.

Gerlee, P., Lundh, T. (2010b) Rock-scissor-paper dynamics in a digital ecology. *Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pp. 285–295.

Gokhale, C.S., Traulsen, A. (2010) Evolutionary games in the multiverse. *PNAS*, vol. 107, no. 12, pp. 5500–5504, DOI:10.1073/pnas.0912214107.

Hallam, S.J., Putnam, N., Preston, C.M., Detter, J.C., Rokhsar, D., Richardson, P.M., DeLong, E.F. (2004) Reverse methanogenesis: Testing the hypothesis with environmental genomics. vol. 305, no. 5689, pp. 1457–1462, DOI:10.1126/science.1100025.

Hamilton, W.D. (1964) The genetical evolution of social behaviour. I. *Journal of Theoretical Biology*, vol. 7, no. 1, pp. 1–16, DOI:10.1016/0022-5193(64)90038-4.

Katsuyama, C., Nakaoka, S., Takeuchi, Y., Tago, K., Hayatsu, M., Kato, K. (2009) Complementary cooperation between two syntrophic bacteria in pesticide degradation. *Journal of Theoretical Biology*, vol. 256, no. 4, pp. 644–654, DOI:doi:10.1016/j.jtbi.2008.10.024.

Kircher, J.W. (2002) The gaia hypothesis: fact, theory, and wishful thinking. *Climatic Ghange*, vol. 52, no. 4, pp. 391–408, DOI:10.1023/A:1014237331082.

Lovelock, J.E. (1972) Gaia as seen through the atmosphere. *Atmospheric Enviroment*, vol. 6, pp. 579–580.

Maynard Smith, J. (1978) Optimization theory in evolution. *Annual Review of Ecology and Systematics*, vol. 9, pp. 31–56, DOI:10.1146/annurev.es.09.110178.000335.

Maynard Smith, J. (1982) Evolution and the theory of games. Cambride, UK: Cambridge University Press.

Nash, J. (1950) Equilibrium points in n–person games. *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48–49.

Nowak, M.A. (2006) Five Rules for the Evolution of Cooperation. *Science*, vol. 314, no. 5805, pp. 1560–1563, DOI:10.1126/science.1133755.

Pfeiffer, T., Bonhoeffer, S. (2004) Evolution of cross-feeding in microbial populations. *Am Nat.*, vol. 163, no. 6, pp. E162–35.

Philips, J.D. (2008) Goal functions in ecosystem and biosphere evolution. *Progress in Physical Geography*, vol. 32, no. 1, pp. 51–64, DOI:10.1177/0309133308089497.

Rawlings, D.R., Johnson, D.B. (2007) The microbiology of biomining: development and optimization of mineral-oxidizing microbial consortia. *Microbiology*, vol. 153, no. 2, pp. 315–324, DOI:10.1099/mic.0.2006/001206-0.

Schelling, T.C. (1980) The strategy of conflict. Harvard University Press.

Shannon, C.E. (1948) A mathematical theory of communication. *The Bell System Technical Journal*, vol. 27.

Stams, Alfons J. M. (1994) Metabolic interactions between anaerobic bacteria in methanogenic environments. *Antonie van Leeuwenhoek*, vol. 66, no. 1, pp. 271–294, DOI:10.1007/BF00871644.

Ulam, S. (1962) On some mathematical problems connected with patterns of growth of figures. *Essays on Cellular Automata*.

von Neumann, J. (1928) Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, vol. 100, no. 1, pp. 295–320.

von Neumann, J., Burks, A.W. (1966) Theory of self–reproducing automata.

Watson, A.J., Lovelock, J.E. (1983) Biological homeostasis of the global environment: the parable of Daisyworld. *Tellus B*, vol. 35B, no. 4, pp. 284–289, DOI:10.1111/j.1600-0889.1983.tb00031.x.

Weissten, E. (2011) "linear stability.". From MathWorld–A Wolfram Web Resource., accessed: 2011-06-06 `http://mathworld.wolfram.com/LinearStability.html`.

Whade, M. (2008) Biologically inspired optimization methods. Southampton, UK: WIT Press.

Wolfram, S. (1984) Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, vol. 10, no. 1-2, pp. 1–35, DOI:doi:10.1016/0167-2789(84)90245-8.