Evolutionary Phenomena in Varied Dynamics

Tiff Lyman

1. Introduction

In 1992, professor of complex systems Kristian Lindgren designed a multi-agent framework for simulating evolution. Different species representing strategies would compete in a game called “iterated prisoner’s dilemma” to earn more offspring in the next generation. Lindgren showed that his program produces evolutionary phenomena such as mass extinctions, symbiotic relationships between strategies, and even a period of “open ended evolution” that was cut short by current limits in processing power at the time.

Despite being an influential paper, only a couple of people have revisited Lindgren’s work. Kyle Harrington built a similar version as an example of increasing complexity being beneficial to coevolution, but neither he nor Lindgren showed that more complex strategies can fairly compete against or coexist with less complex ones (2). In Lindgren’s most famous result shown below, each era is defined by a more complex strategy emerging and dominating the rest. This is contrary to real-world ecosystems where even apex predators must share their environment with other creatures. This raises the question: is it possible to unbias Lindgren’s program enough to promote fairness for less complex strategies?

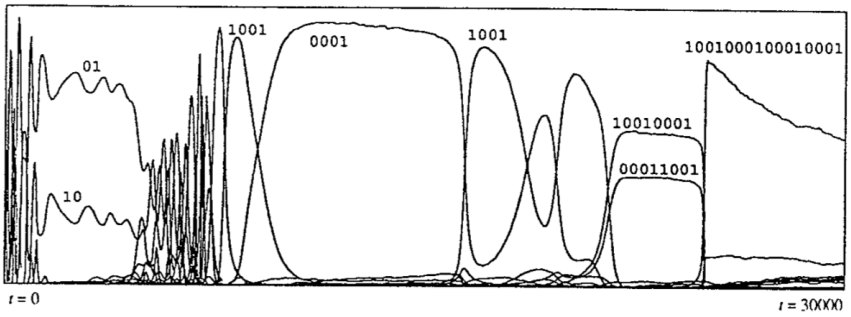

Figure 1: Plot of strategies and their population over time in Lindgren’s simulation, where strategies are represented by a binary number. Each era where one strategy has an overwhelmingly larger population than the rest are consistently caused by a more complex strategy emerging.

2. Background

2.1 Iterated Prisoner's Dilemma

Suppose two people have a stack of money in front of them. Each person can do one of the following: grab the money (defect) or leave the money where it is (cooperate). If neither of them grab it. then they split the money. If only one person grabs it, that person takes all of it, leaving their opponent with nothing. If both of them grab the money, they each only get a dollar from the stack. This is what's called the prisoner's dilemma. By repeating this face-off and adding up the reward from each round, we create an iterated version where opponents can remember what their opponent did previously and use it to their advantage. Some strategies encourage cooperation while others are more hostile or deceptive.

In Lindgren's case, he used the simple payout system shown on the right. Here, a binary value represents a player's move: 0 to defect and 1 to cooperate. Using the same values, we can represent the game's payout as the $p = \begin{pmatrix} 1 & 0 \\ 5 & 3 \end{pmatrix}$ matrix. Each position [i,j] represents a player's reward when they play i and the opponent plays j. In our simulation, we also include noise in the game. When a player goes to declare 0 or 1, there is a small chance that they accidentally do the opposite. We'll explore this idea further in section 3.2.

		Player 2	
		Cooperate	Defect
Player 1	Cooperate	(3, 3)	(0, 5)
	Defect	(5, 0)	(1, 1)

2.2 Genomes

Each player has a memory of size m_i, where m is the number of moves it can remember. For example, a memory of size 1 contains the opponent's last move, while memory of size 2 holds the opponent's last move and the player's last move. This memory can be represented as a binary number, which means every possible history of moves can be indexed between 0 and 2^m.

This allows us to represent a player's strategy as a genome $\mathbf{G = \{A_0, A_1, \ldots, A_n\}}$. This genome is a binary string of length $n = 2^m$, where A_i represents the move to make (0 = defect, 1 = cooperate) when its memory is index i. For example, the memory 1 strategy "tit-for-tat" copies it's opponent's last move with the genome "01".

These genomes increase in complexity through mutations. There are three possible mutations that could occur at the end of a generation.

- **Point Mutation**: For each character of each genome, there is a 1/50,000 chance of the character flipping (ex. 01 => 11)
- **Duplication**: For each genome, there is a 1/100,000 chance of creating a copy of itself and appending it (ex. 01 => 0101). This increases the memory size by 1.
- **Split Mutation**: For each genome, there is a 1/100,000 chance of randomly removing half of itself (ex. 0001 => 00). This decreases the memory size by 1.

2.3 Evolutionary Stable States

Suppose every individual in our simulation is of genotype G. The strategy described is considered *evolutionary stable* if any smaller invading strategy G' dies out before G, meaning G must effectively last forever. This idea, first proposed in 1973 by John Maynard Smith, has only appeared in Lindgren's demonstrations when more memory wasn't an option. As shown earlier in figure 1, the largest memory genomes that Lindgren allowed were of size 4. These states aren't perfect. Boyd + Loberbaum proved that there are no pure strategies for IPD, but the odds of stability breaking in those cases are insignificant for the scope of this experiment (3).

3. Experiments

3.1: Population Dynamics

In this setup we have a population of N = 10,000 where each genome's population is represented as a fraction of that total. We initialize the environment with each of the four memory 1 genomes (00, 01, 10, 11) set at a population of 1/4. For genome i, let $x_i(t)$ represent the fraction of the population with genotype i at generation t. To calculate a genome's score s_i, we use the following equation

$$s_i = \sum_j g_{ij} x_j$$

where g_{ij} is the average number of points earned by genome i when playing against genome j. Originally Lindgren used a Markov matrix to play games and used scores relative to the average as fitness. With modern computing ability, I thought it would be fun to actually have genomes play the game and use the following logistic equation presented in Lindgren's later work (4).

$$x_i(t+1) = x_i(t) + ds_i x_i(t)\left(1 - \frac{\sum_j s_j x_j(t)}{s_i}\right)$$

In this equation, the value d is a growth constant (d = 0.1) and population size is theoretically constant. The only time where it's not constant is when a genome dies out. When a genome's population dips below 1/N, it is removed from the population. To counteract this, we equally distribute what was left of the dead population among those still present.

3.2: Noise

Previously, there was always a 1/100 chance of a genome accidentally playing the move opposite from what it intended. Our goal in this section is to induce more noise as genomes become larger and use instability as a handicap. To do this, we make the probability of noise proportional to genome size. For a genome of length 2^m, we say the new odds are $2^m/200$ that the genome experiences noise. This maintains the same amount of noise for all of the memory 1 genomes, but limits larger genomes' ability to follow specific patterns. We ran the simulation for 30,000 generations with the noise handicap and obtained the following results.

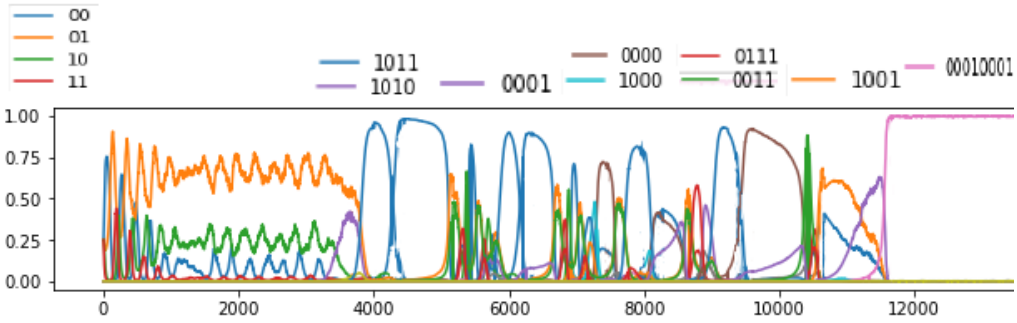

Figure 2: Plot of genomes when affected by a noise handicap. Labels are placed approximately where that genome first appeared, but there is some overlap from repeating colors. No significant changes occurred beyond t=12,000

Contrary to Lindgren's 1992 run where four temporary eras of stability were present, here we only see two periods of stability with a chaotic series of interactions in between. What hasn't changed is that each new period is caused by a larger genome dominating, but that doesn't mean there aren't some interesting takeaways. Despite having a highly noise-resilient memory 3 genome dominate for the last 18,000 generations, a higher memory genome was never capable of

emerging. One possible explanation is that the probability of noise in a memory 4 genome (1/12.5) is greater than the probability of an arbitrary index being acted on (1/16).

3.3: Mutations

Next, we aimed to promote diversity before increasing in memory size. This is done by adjusting the probability of each mutation. For point mutations, we simply double the chance of inducing a flip. For duplications and splits, we again integrate memory size. For a genome of memory size m, the probabilities are as follows:

- Duplication chance: 1 / (m * 100,000)
- Split chance: m / 100,000

We ran this for 30,000 generations (with the original amount of noise at 1/100) and obtained the following results.

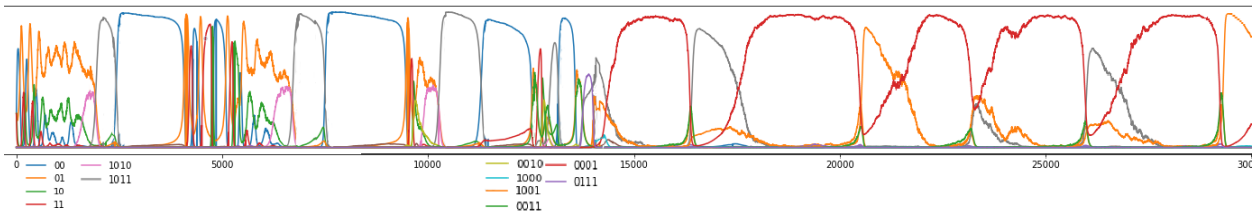

Figure 2: Plot of genomes when affected by a mutation handicap. Despite not appearing on the graph, some higher memory genomes emerged around t=13,000 but none grew enough to appear on the graph

Even after 30,000 generations, no genomes greater than memory 2 rose to the top. Despite this limiting the genome pool, each of the drastic shifts in population come with unique properties. For example, the genome 0001, which cooperates until their opponent defects and then only defects, has a few waves of popularity but its main competitor switches between instances. Additionally, all memory 2 genomes appear at some point with half of them becoming popular enough to be clearly visible on the graph.

3.4: Size Balancing

One last option we thought might work would be to apply a handicap in the logistic equation, but this requires ensuring that total population stays constant. Similar to how relative

score is calculated, we can scale the equation with the genome's relative memory. For genome i with memory m_i the equation is as follows:

$$x_i(t+1) = x_i(t) + ds_i x_i(t)\left(1 - \frac{\sum_j s_j x_j(t)}{s_i}\right)\left(\frac{\sum_j m_j x_j(t)}{m_i}\right)$$

This equation should bias slightly towards conformity, so that there has to be a remarkable breakthrough to increase in complexity. The simulation was run with all mutation and noise probabilities reset to their defaults and ran for 30,000 generations.

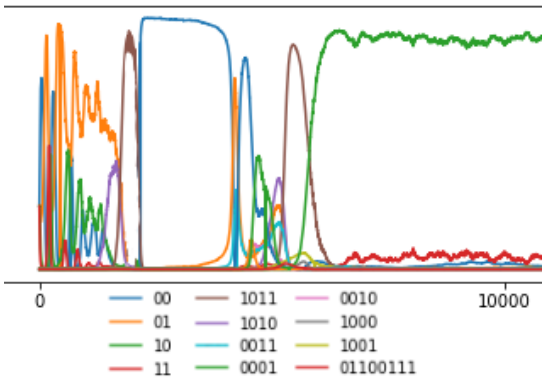

Figure 3: Plot of genomes when affected by a logistic handicap. This run went faster than usual but resulted in a less complex genome developing a coexistence with a secondary, more complex genome.

The memory scaling may have universally sped up evolution because a relatively stable state was reached within just 7000 generations. This stable state is especially remarkable because it consists of a dominant memory 2 genome, 0001, and a secondary memory-3 genome population, 01100111. The smaller population is clearly nonzero so there must be some explanation for why such a small group was able to coexist for the remaining 23,000 generations.

4. Discussion + Future Work

Obtaining a result where a smaller genome quickly dominates even more complex genomes is

nice, but it feels forced. Then again this entire system is man-made so it's hard to call it perfect, but revisiting it has provided some new insights. In the original paper we only saw two memory 2 genomes which were quickly overpowered. Under similar conditions and similar duration, our results ended with a smaller genome on top in the majority of runs and over four times as many

memory 2 genomes with significant populations. Even those that were present both in Lindgren's version and this one performed differently. For example, 1001 was never the top genome in the third run and 0001 was "evolutionary stable" by our standards.

Genome	1	2	3
0000	✔		
0001	✔	✔	✔
0010		✔	✔
0011	✔	✔	✔
0111	✔	✔	✔
1000		✔	
1001	✔	✔	✔
1010	✔	✔	✔
1011	✔	✔	✔

Table 1: Chart of memory 2 genomes by run in which they appeared and grew significantly in population. We define this by making up over 5% of the population.

Now that there exists code available for experimentation, there's lots more that can be done with Lindgren's system. With more time I would like to explore signaling and allowing opponents to tell the opponent what they're about to do. This would allow genomes to explore honesty versus deception. Having the agents actually play the game instead of using a markov chain might make this easier, but either way it's more than I can do in this semester alone. Another idea would be to instead use a minimum criterion similar to Brant and Stanley, which would likely encourage diversity and speciation at higher memories.

References:

1. Lindgren, Kristian. (1991). Evolutionary Phenomena in Simple Dynamics. Artificial Life II.

2. Kyle Harrington, Jordan Pollack; Escalation of Memory Length in Finite Populations. *Artif Life* 2019; 25 (1): 22–32. doi: https://doi.org/10.1162/artl_a_00278

3. Boyd, R., Lorberbaum, J. No pure strategy is evolutionarily stable in the repeated Prisoner's Dilemma game. *Nature* 327, 58–59 (1987). https://doi.org/10.1038/327058a0

4. Kristian Lindgren, Mats G. Nordahl, Evolutionary dynamics of spatial games, Physica D: Nonlinear Phenomena, Volume 75, Issues 1–3, 1994, Pages 292-309, ISSN 0167-2789, https://doi.org/10.1016/0167-2789(94)90289-5.

5. Brant, Jonathan & Stanley, Kenneth. (2017). Minimal criterion coevolution: a new approach to open-ended search. 67-74. 10.1145/3071178.3071186.