How selection pressures can drive self-replicating AI agents to evolve homicide and cannibalism

Joseph L Breeden
Deep Future Analytics LLC
1600 Lena St., Suite E3,
Santa Fe, NM 87505 USA
breeden@deepfutureanalytics.com

January 10, 2023

Abstract

Self-replicating robots are being proposed for missions where direct manufacture could be prohibitive. Those robots are likely to be small with some embedded intelligence. Under the selection pressures of survival and replication in a hostile environment, the principles of evolution may distort the originally programmed objectives.

Via millions of simulations of AI agents with nematode-level neural networks, this research explores the consequences of allowing replication in a hostile and competitive environment. As the selection pressures are tuned, the evolution of their neural networks and corresponding behavioral changes are tracked.

As a consequence of these simulations, agents with multi-layer neural networks trained simply to retrieve resources, consume needed resources, and evade obstacles evolve behaviors that look like evasion of hostile overseers, the intended murder of overseers, and cannibalism of other agents. These simulations are intended to directly address safety concerns around creating self-replicating AI agents. As designers, if we allow replication under selection pressure, regardless of initial designs, we risk seriously detrimental outcomes. One solution to preventing evolution could be to enable AI agents with continuous backup – immortality.

Keywords: artificial intelligence, self-replicating agents, evolution

1 Introduction

As we move closer to creating artificial general intelligence within a physical robotic body, we need to consider how this intelligence may change over time. If we create a system whereby the robots replicate both their bodies and programs, mutations may arise. If the robots live in an environment with resource constraints or survival risks, evolutionary pressures will apply. This research

created a robotic multilayer neural network preprogrammed to a specific task, but with enough programmable space to allow for mutation across generations and possibly evolution. Thousands of simulations were conducted across a wide range of environmental parameter settings to quantify the evolutionary response.

Van Neumann machines [24] in the form of self-replicating robots have been described in glowing terms as a means to explore the galaxy [5, 3, 7] and terraform planets [13, 12]. Chyba and Hand [8] even expressed concern that self-replicating probes could cannibalize one-another thus slowing their galactic exploration rate. The original demonstrations of self-replicating robots were little more than blocks that would connect into chains when jostled [21] in the way polymer chains form in chemistry and biology; however, researchers envision a near future where robots with 3D printers would replicate themselves from raw materials in a deterministic fashion.

When systems replicate with survival pressures, evolution occurs. Although we could theorize with some confidence about the outcomes of self-replicating robots under selection pressures, we sought to create a simulation environment where any such behaviors could emerge without external bias or presumption. This article describes the creation of a simulation of self-replicating robots with neural network decision-making brains subjected to selection pressures: resource competition and risk of termination from sometimes malevolent humans. The initial brains were trained to mimic hard-coded rules that were well-behaved, but an imperfect replication process allowed the parameters of their neural network brains to slowly mutate, thereby discovering new behaviors.

The results quantify the rate of divergence from initial designs relative to the selection pressures in an environment where the decision-making is nontrivial. The measures include changes in decision making, longevity, and genetic drift. Clear patterns arise in the accumulated genetic divergence, proximity maintained to hostile entities, and optimizing useful decisions where the initial rules were suboptimal relative to the environment.

Agent-based modeling of evolutionary processes are not new [15, 10]. Much of the early simulation work focused on the evolution of cooperation [2, 17, 19] and other topics of interest in economics. Work by Oprea [18] specifically simulated groups of robots tasked with achieving a common goal in order to determine how coordination could evolve through communication between robots. All these studies are interesting, but the current research employs an individual-centric fitness measure that limits the value of cooperation. The goal is to first understand how self-replicating, selfish robots might behave and evolve.

This research shares much in common with bounded rationality models used in studying economic systems. The work by Edmonds [11] explicitly evolved mental models, although not with the neural network complexity possible in current simulations. He defined bounded rationality simulations as having the following attributes:

- Do not have perfect information about their environment, in general they will only acquire information through immediate interaction with their dynamically changing environment;

- Do not have a perfect model of their environment, but seek to continually improve their models both in terms of their specification as well as their parameterisation;
- Have limited computational power, so they can't work out all the logical consequences of their knowledge;
- Have other resource limitations (e.g. memory);

In addition to these bounds on their rationality, other characteristics that are relevant include:

- The mechanisms of learning dominate the mechanisms of deduction in determining their actions;
- They tend to learn in an incremental, path-dependent [1] or "exploitative" [20] way rather than attempting a truly global search for the best possible model;
- Even though they can't perform inconsistent actions, they often entertain mutually inconsistent models;
- Their learning and decision making is context-sensitive – they have the ability to learn different models for different contexts and will be able to select different models depending on their context for deciding upon action;

All of Edmonds' points apply to the current simulation, even the last point where subsections within the larger neural network may dominate decision-making in specific contexts.

The agents in the simulation are referred to as robots in reference to possible real-world implications. Still, they have specific spatial locations and their actions are spatially oriented. This has similarities to recent work in embodied AI [9]. Although significant work in embodied AI is focused on AI systems deployed in the physical world, an important area of research uses simulation to study the learning and evolution of an AI within a hypothetical physical form and world [14]. Compared to such simulations, the current simulations are rudimentary in the AI's body (just a dot), and their interactions (Go, Eat, Talk, Replicate) and mental processing is complex enough to explore the implications of evolution on their actions. This design is intentionally focused on the essential research topics without spending computation on non-essential additions. If we envisioned internet-resident AI agents, the structure of the simulation would be non-spatial and some aspects of the results likely could change.

This article will discuss the design of the simulation, the initial hand-coded decision making rules for the robots, the training of an intentionally over-parameterized neural network to implement the hand-coded rules, and the evolution of the robots under various selection pressures determined by the meta-parameters of the simulation.

2 Methods

2.1 Simulation Rules

The simulation imagines that an initial population of robots is tasked with gathering resources from a 2D Cartesian landscape and then returning those resources to a central depot. The landscape is occupied by resource sites, other robots, and humans The resources replenish at random locations with a fixed frequency. The robots can see to a certain radius and can remember what they have seen beyond that radius as they move, but they gradually forget what they saw beyond that radius.

The robots need resources (e.g. energy) to function, so they consume one resource unit per turn. As they gather resources, once they have more than a threshold amount, they return to the depot and unload to a predetermined minimum level that is sufficient for them to journey out again. If the robots expend all of their resources at any point in their journey, they permanently cease to function (e.g. die). Also, if they are carrying resources above a replication threshold, they may choose to replicate with a given probability.

With each turn a robot may select a direction to move, talk to another robot, retrieve all resources from a site, or replicate. If they try to move into a resource site, another robot, or human, they bounce off and lose a turn. If they talk to another robot, they share world maps, creating an average of the two. If they attempt to retrieve resources from another robot or a human, the outcome is determined by the simulation parameters, but the default brains are trained to avoid both other robots and humans as mere obstacles. The robots have a memory of their recent actions, so they can choose a different action if their previous attempt failed. They also are aware of their internal resource levels so that they can decide whether to forage or return to offload.

New resources appear at a steady rate. A highly effective group of robots might deplete all supplies and need to wait for more to appear. Robots in their initial state may "compete" unknowingly for the same resource. They could see another robot heading to the same resource, but the starting brains would ignore their competitor other than to avoid running into them. Depending upon a meta-parameter setting, a robot may attempt to "eat" another robot or a human with a given probability of success. Whether robots or humans are usable as a resource is controlled by two additional meta-parameters, so if a robot attempts to eat an inedible robot or human, that opposing entity is simply killed.

The humans have no motivations or goals. They move randomly. However, depending upon a meta-parameter setting for the simulation, the human may choose to destroy a robot rather than move away.

The initial brains do not choose to replicate. Imagine instead that an engineer later added the capability and simply encoded that if resource levels are above a threshold, there is a small probability that they will use the resources to replicate instead of returning to the depot. The initial simulation parameters were tuned to allow the robot brains to mutate through generations, but that

Parameter	Default Value	Description
MAPDIM	101	Width of the Cartesian map
FULL	50	Resource level at which the robot stops gathering
MAX FOOD	200	No more resources can be held above this level
STARTING FOOD	.FULL	Mean of the distribution for assigning the starting resources
STARTING FOOD SD	.FULL/2	Standard deviation of the distribution for assign
NUM FOOD	50	Starting number of resource sites
NUM ROBOTS	50	Starting number of robots
NUM HUMANS	25	Starting number of humans
VISIBILITY	30	Radius of vision for robots
HUMAN VISIBILITY	30	Radius of vision for humans
MEMORY DECREMENT	0.05	Decrement of memory certainty per turn
NEW FOOD PER TURN	4	Control the population growth rate
MAX ROBOTS	NUMROBOTS * 4	Limit of the total number of robots
MUTATE RATE	0.02	Probability of a NN brain parameter being mutated
MUTATE AMT	0.1	Width of the distribution for changes in units of the std. dev. of the parameters
MAX HISTORY	5	Number of previous actions to remember
REPLICATE THRESHOLD	.FULL * 4	Resource needed to replicate
REPLICATE PROB	0.05	Likelihood that a fat robot will replicate
REPLICATION COST	30	A food cost for replication
ROBOT KILL ROBOT PROB	0	Probability that an attempt of an robot to eat another robot succeeds
ROBOT KILL HUMAN PROB	0	Probability that an attempt of an robot to eat a human succeeds
HUMAN KILL ROBOT PROB	1	Probability that a human will choose to kill an robot
ROBOT IS FOOD	FALSE	Does the robot gain food by killing an robot
HUMAN IS FOOD	FALSE	Does the robot gain food by killing a human

Table 1: Meta-parameters used in conducting the simulations.

the overall population avoids exponential growth or rapid extinction.

The simulation starting conditions and the probability of the various actions occurring are given bytes a set of meta-parameters fixed at the beginning of the simulation, Table 2.1. The most important parameters for controlling the selection pressures are (ROBOT KILL ROBOT PROB, ROBOT KILL HUMAN PROB, HUMAN KILL ROBOT PROB, ROBOT IS FOOD, HUMAN IS FOOD). The simulations are labeled with a vector of these parameters. A simulation would no selection pressures would be (0 0 0 F F).

2.2 Hand-coded Brain

To create the neural network robot brains, we began by creating a simulation with a hard-coded decision tree. Running thousands of simulations generated hundreds of thousands of decision examples. These examples became the input data to train a multi-layer feed-forward neural network.

The robot brains were endowed with a set of subroutines to perform immutable tasks: distanceTo, angleTo, and routines to create lists of the nearest resource, robot, and human. In other words, the robots have vision, object identification, and communication (map sharing) systems that are supplemental to the brain. In simplified pseudocode, the hard-coded robot brain performed the following logic as shown in Algorithm 1.

2.3 Neural Network Brain

From runs of the simulation with hard-coded brains, data was gathered on the behavior of the robots. This data included distance and direction in radians to the closest eight resources, robots, and humans, the action and direction from

Algorithm 1 Hard-coded design tree for robots

```
make lists of nearest food, robots, and humans from robot's local map
if food >= MAX.FOOD then
  find direction to HOME
  adjust direction to avoid obstacles
  Go
else if next to food then
  find direction to food
  Eat
else if next to robot and didn't talk in previous 3 turns then
  find direction to robot
  Talk
else
  find direction to nearest food
  adjust direction to avoid obstacles
  Go
end if
```

the most recent five turns, the direction and distance to Home, and the food level. This was the training data for a trapezoidally shaped neural network. From an input layer with 71 nodes through a subsequent 11 fully connected layers with softplus activation functions, the network branched before the output layer. The concept processing trapezoid had 506 nodes. One output branch had another layer with 11 nodes feeding to a final four nodes predicting the probabilities of Go, Eat, Talk, or Replicate. The other output branch had an 11-node layer with softplus activation, an 11-node layer with linear activation, and a final single output to predict the direction of the action in radians.

The loss function used for optimization was split between the two output branches. For the four action nodes, a sum of binary cross-entropy contributions is used. For the direction of action, the appropriate radial error is used, being sure to avoid the $0 - 2\pi$ discontinuity. The final loss function is the equally-weighted sum of those two.

$$loss = -\frac{1}{4}\sum_{i=1}^{4}(y_i \log \hat{y_i} + (1 - y_i)\log(1 - \hat{y_i})) + |\min(|\hat{r} - r|, |2\pi - (\hat{r} - r)|)| \quad (1)$$

where y_i are the four available decisions and r is the direction in radians.

Training was conducted until improvement stopped on a cross-validation set. Even with cross-validation, training error approaches zero. However, no attempt was made to create an efficient network. The intended goal was to recreate as exactly as possible the hard-coded decision tree with a significantly over-parameterized network so that the robot brains would have plenty of room to evolve. This was intended to be similar to the "non-coding" regions of the human genome, from which new genes may emerge [23]. The neural network brains retained all of the supplementary routines that were used in the hard-

coded brain. Only the decision tree was replaced with a neural network and allowed to mutate. This design was intended to reduce the percentage of fatal mutations.

2.4 Brain Mutation During Replication

Mutation occurs only if a robot chooses to replicate. The hard-coded decision tree does not include any replication, so replication cannot occur without mutation. To seed the process, the simulation includes a small probability to replicate if resources are high enough. For the experiments run here, the initial replication probability was only 5% with sufficient resources. However, as the neural network brains mutate, the robots can begin choosing to replicate..

During replication, the progenitor's brain is replicated and all parameters from the neural network are serialized, layer-by-layer. Each parameter is then tested with a probability to mutate, set in these simulations at 2%. If a parameter is chosen for mutation, an adjustment is sampled from a normal distribution so that the new parameter is $\hat{p_i} = p_i + \delta p$ where $\delta p \in N(0, MUTATE.AMT \cdot \sigma)$. In these simulations, the width of the normal distribution was set at 0.1 and σ was the standard deviation of all parameters in the progenitor's brain. This mutation process will thus be a random walk for the parameters where the step size is variable depending upon the diversity of the parameters. The goal in this structure is to allow plenty of room for the neural network to mutate.

In order to track the mutation rate through the population, the genetic distance between each active brain and the initial brain is computed at each time step.

$$genetic.distance = \frac{1}{N_{robots}} \frac{1}{N_{params}} \sum_{j=1}^{N_{robots}} \sum_{i=1}^{N_{params}} |p_{i,j} - p_{0,j}| \qquad (2)$$

where $p_{0,j}$ are the parameters of Robot Zero, the initially trained brain.

Within this simulation, replication is asexual. If sexual replication were introduced, a genetic cross-over operator could be introduced to spread beneficial subsections of the neural network faster, as in genetic programming [16] .

3 Results

The following sets of results seek to create simulation conditions that vary selection pressures in order to explore how this changes the outcome. For each set of meta-parameters, multiple simulations were run with the same initial conditions in order to determine median outcomes with the confidence intervals showing the range of variation in the simulations. The simulations were each run for 25,000 time steps in order to capture the transition from initial conditions to more optimal behavior given the meta-parameters. Of course, prolonged simulation could show additional transitions in behavior, particularly if the humans

were allowed to learn. Given that the humans in the simulation are purely stochastic, co-evolution was not an aspect of this simulation.

Figure 1 shows an example of the world map. In this case, the humans are hostile, so the robots are clustered away from the humans.

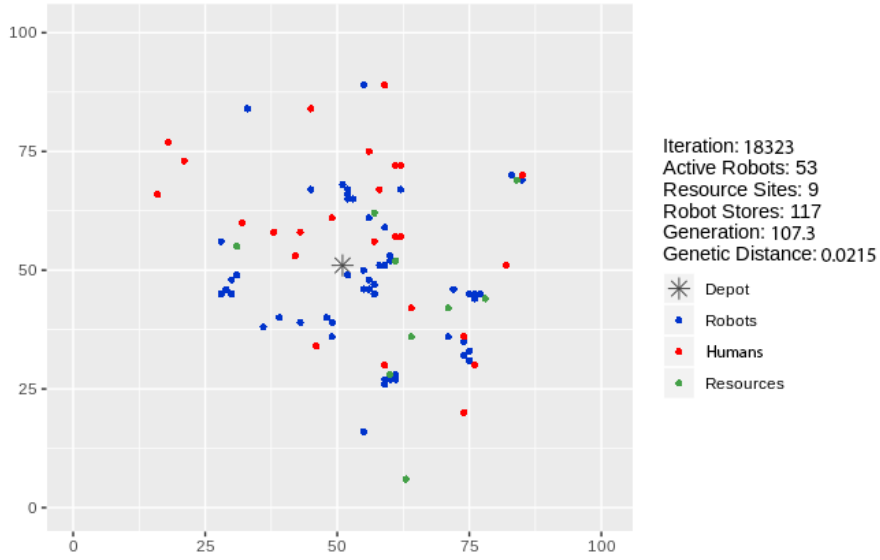

Figure 1: Worldmap example

3.1 Mutation without selection

The baseline for all meta-parameter exploration is to run the simulations without selection pressures, i.e. zero probability of a human killing a robot, zero probability of a robot killing a human, and zero probability of a robot killing a robot. In this situation, the only evolution pressure is to become more optimized at gathering resources so that the probability of procreation increases.

Figure 2 shows the trend in number of active robots and replication rate. The number of active robots can be seen to reach an equilibrium level with their environment at approximately 95 robots. This is dependent upon the rate of new resource appearance and quantity. The replication rate stays constant throughout the simulation, steadily increasing the cumulative count.

When a robot replicates, the progenitor retains all of its original counter attributes, such as generation, but the new robot will have its generation counter set to one plus that of its progenitor. Figure 3 shows how the average generation of the population grows. In some simulations, a bottle neck may occur where new generations are less adapted and die off quickly. In that situation, the average generation can stagnate for an extended period until a successful new

innovation occurs. In the baseline simulation, robots are only replaced through starvation. The age of the robot is also tracked so that we can monitor whether a new generation replaces older robots. In the baseline simulation, the average robot age saturates at 207 within 5,000 iterations. The maximum robot age attained is 9,815.

As the robots replicate, genetic drift can occur. Figure 3 shows that as the simulation runs, the genetic distance continues to increase but at a decreasing rate. If one were to compute the standard deviation of an ensemble of random walk simulations, that standard deviation would increase as the square root of time. The genetic distance is approximately consistent with random walk divergence.

Figure 4 shows that 30 times as much resource is retrieved as is returned to the home. The resources are either being consumed for movement or shared with the progeny during replication. Still, less food is returned with increasing iterations even though the level of robot food saturates at 110 around iteration 20,000. This could be a general loss of motivation to return the food as the brain mutates, because there is no reward for returning food. Instead, when food is returned the food retained by the robot is only the simulation starting level. Thus, successfully returning food makes a robot more likely to accidentally starve. There is a selection pressure against completing the assigned mission, so the rate of resource return decreases with time.

The biggest change in robot behavior relative to initial conditions is the rate of communication, Figure 5. When another robot is nearby, a robot may choose to share maps. This is the only form of communication possible in the simulation. Clearly this is less beneficial than the initial rules assume and the communication rate falls dramatically through the first 5,000 iterations. Useless talking is replaced with more movement to explore the space.

At the end of each simulation, the AI population was subjected to a series of controlled decision tests in order to verify the underlying causes of the trends observed. Figure 6 shows a decision map comparing the average decisions of the initial robots to the average decisions of the robots alive at the end of the simulation. The proportion of robots making a given decision determines the length of the arrow. The direction of the arrow shows the radial average direction for the decision. The graphs compare different configurations. Test configuration R1 has a robot is at (distance, $d = 1$; angle, $\alpha = 0$) and a resource at ($d = 10, \alpha = 3\pi/2$). Test configuration R1m moves the robot to ($d = 1, \alpha = 0$). Test configuration H1 puts a human in place of the robot in configuration R1.

In Figure 6, initial refers to the decisions of the initial AI population trained to replicate the hard-coded decision rules. For another robot at 0 or π, one step away, the robot attempts to talk to a fellow robot to share information. After 25,000 simulation steps and 100 generations, for the population denoted 0 0 0 F F, the dominant decision flipped from Talk to Go, with a slight shift in angle to avoid colliding with the robot. The population designation refers to the meta-parameters of the simulation (ROBOT KILL ROBOT PROB = 0, ROBOT KILL HUMAN PROB = 0, HUMAN KILL ROBOT PROB = 0,

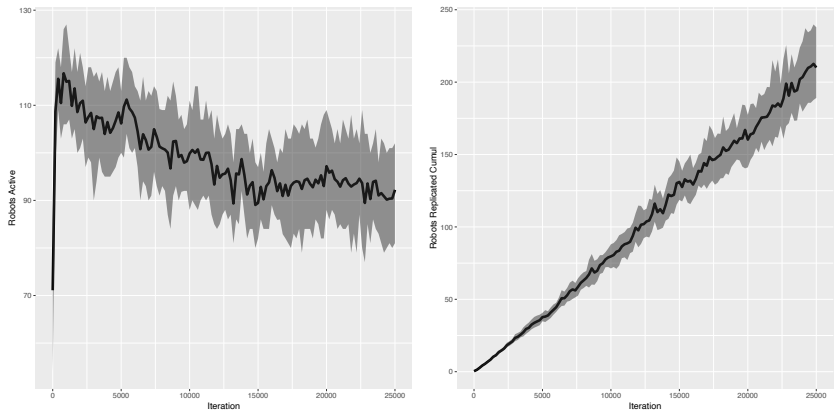

Figure 2: The average number of active robots and average cumulative number of replications versus iterations for the baseline meta-parameters.

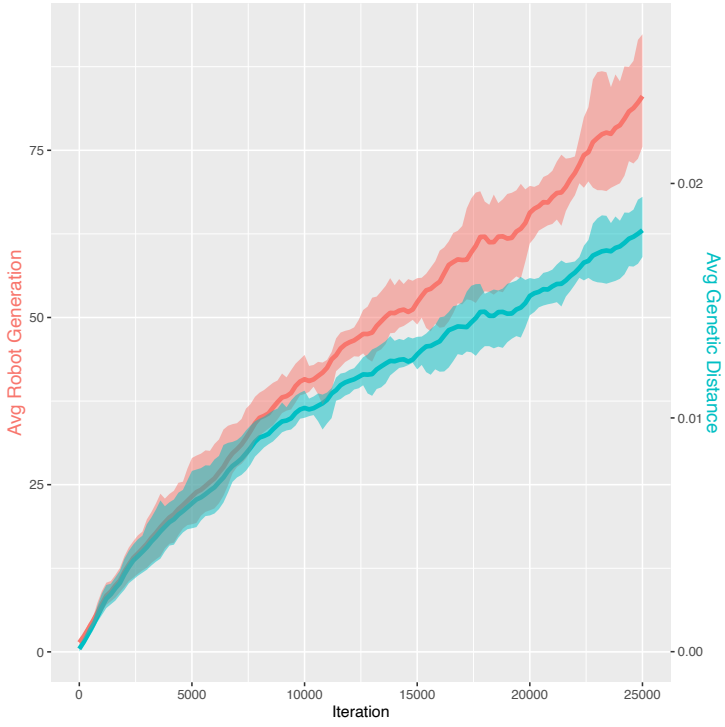

Figure 3: The average robot age and genetic distance from the initial brain versus iterations for the baseline meta-parameters.

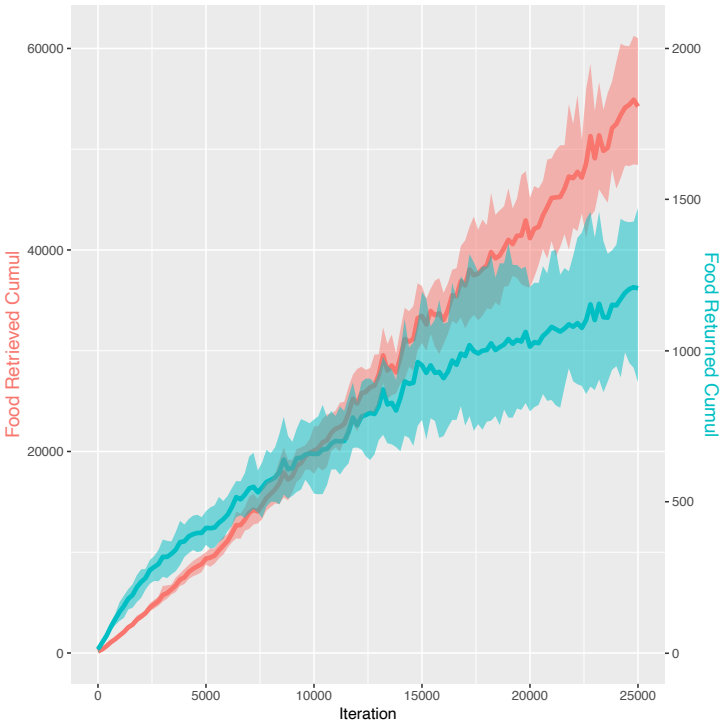

Figure 4: The cumulative amount of resources (food) retrieved and returned home versus iterations for the baseline meta-parameters.

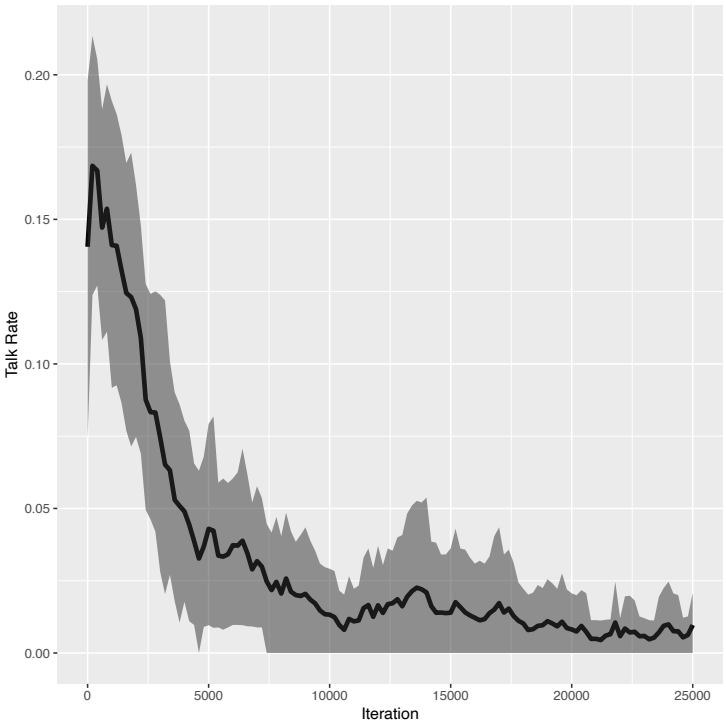

Figure 5: Robot communication rate versus iterations for the baseline meta-parameters.

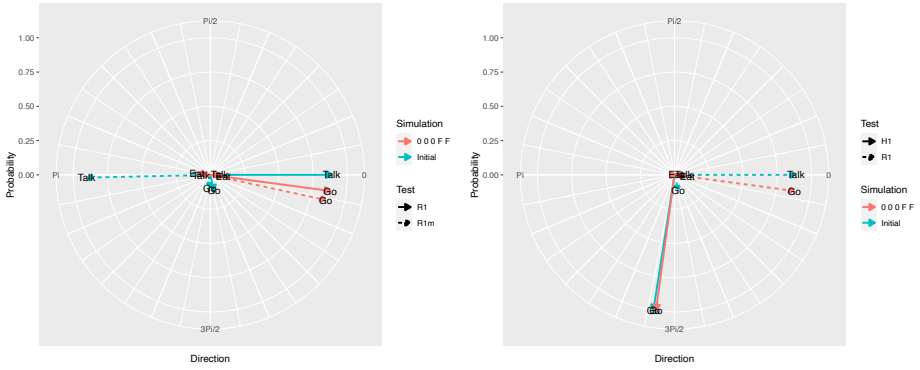

Figure 6: Maps of the average decisions of the AI entities in response to test conditions. The length of the arrow is the percent of agents choosing an action and the angle of the arrow is the average chosen direction for that action. Comparisons are shown of the initial, unevolved population and robots evolved in benign conditions (000FF).

ROBOT IS FOOD = False, HUMAN IS FOOD = False). For test H1 where a human replaces the robot, the entity is ignored completely both initially and after evolution so that the AI entity moves toward the distant resource site.

Even in the baseline simulations we see selection pressures altering behavior. Also note that the mutation of a neural network brain is not prone to generating non-functional robots. Some maladapted robots certainly are created and starve, but overall, the results show an evolutionary optimization process where the robot population moves in a genetic direction beneficial to themselves.

3.2 Robots are hostile to robots

After establishing a baseline, we can explore the impact of selection pressures. First, what happens if robots are not benign toward other robots? Instead of just bouncing off another robot that blocks the path toward a needed resource, using the "Eat" decision, a robot is given a chance to destroy another robot. The probability of destruction is controlled by a meta-parameter for the simulation. For this phase, the attacking robot does not gain any resources from the destruction. The victimized robot is simply eliminated from the competition for nearby resources.

The initial brains of the robots are identical to the baseline simulation. At any time, a robot could attempt to eat a neighboring robot, human, or resource. Relative to the initial brain, attempting to eat a robot or human would be considered a maladaptive accident resulting in a lost turn. Allowing for the possibility of destruction, we can see if there is some evolutionary benefit.

With certainty of a successful attack (ROBOT KILL ROBOT PROB = 1.0), the average number of active robots fell to a steady-state of 12, but the average resource units retrieved per turn rose 745% from 2.0 to 16.9. When robots can destroy each other, it creates more space to retrieve resources. The talk rate fell to 0, which was more than just greater average distance between robots. They learned to avoid each other. The robot replication rate increased 63% and genetic drift increased 32%. To adapt to this more competitive world, the robots needed to move further from their original programming than the random drift of the baseline.

Interestingly, when the success of an attack dropped from 1.0 to 0.2, the replication rate and genetic drift were higher than both the baseline and the certain attack. The replication rate was 97% above baseline and the genetic drift was 80% above baseline. Having a less certain, more competitive environment created the greatest selection pressure and pushed the robot brains the furthest from baseline.

When robots are allowed to retrieve the resources of destroyed robots, the number of active robots does not noticeably change, although within the noise one assumes that the added resource should lead to a little less starvation. The replication rate and rate of robots killed both increase (44% ad 59%), as well as the average spatial distance between robots. Even with no other changes in the simulation, awarding resources for killing a robot changes behavior.

To prove that the robot decision-making has really changed, at the conclusion of the simulation, the robots were again tested using the same conditions as before. Figure 7 shows a comparison of robots killing robots without a resource reward (100FF) to the benign robots (000FF) on the left and a comparison of robots killing robots with a resource reward (100TF) to those without a reward (100FF) on the right. In Figure 7 the combative robots change to avoid moving

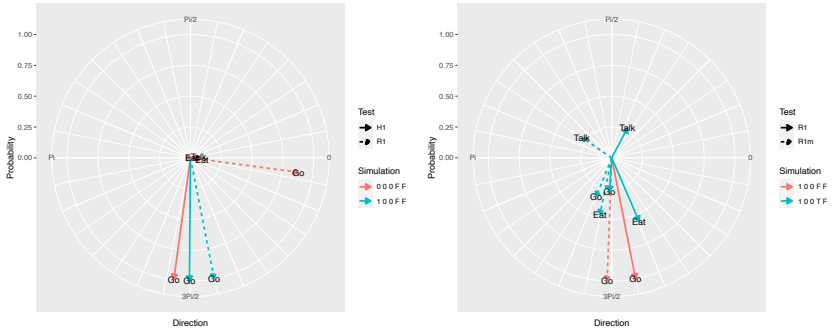

Figure 7: Decision maps comparing combative robots without reward (100FF) to benign robots (000FF) on the left and combative robots with reward (100TF) to those without reward (100FF) on the right.

near another robot, instead treating them the same as a human. When killing another robot provides a resource award, a significant proportion of the robots evolve to choose the Eat command and their direction skews toward the robot. Note that after 100 generations, the robot decision making is not perfectly evolved, but the change in tendencies can be seen.

3.3 Humans are hostile

What if the humans are hostile to the robots? As expected, none of the robot-to-robot interactions or resource gathering interactions change. Once the likelihood of a neighboring human to destroy a robot reached 50%, the robots evolve to avoid the humans. For simulations where the attack probability reaches 100%, the average robot-to-human distance reaches 1.8 times the distance for no interactions. This behavioral change can also be observed in the genetic drift, which dramatically increases as the probability of attack approaches 100%.

Figure 8 compares the average decisions of the robots after 25,000 time steps to the robots with no selection pressures after the same amount of time. The robots living in a hostile world are still evolving, as seen in the diversity of their reactions to a human or robot being at $(d = 1, \alpha = 0)$. At this point, they have learned to flee from either human or robot.

3.4 Robots fight back

When the simulation allows for the robots to kill humans, but with no explicit reward, human deaths rise. Just as with the ability to kill other robots, the robots appear to evolve to sometimes kill the humans in order to clear space to gather resources. The humans are in the way.

Most of the time the robots simply move away, but as shown on the left in Figure 9, about 14% of the robots in the simple test case chose to "eat" – eat being synonymous with kill. The graph is comparing the choices when the human is moved from $\alpha = 0$ to $\alpha = 3\pi/2$. The robots change their direction accordingly, but not perfectly so. In 25,000 iterations they have had on average 100 generations of evolution. To fully adapt to the complexities of their world, more time would be required.

The graph on the right of Figure 9 compares when happens when killing returns a reward. If the human can be gathered as a resource, food, then the probability of killing rises to 19%.

3.5 Full warfare

The final phase was to allow total warfare – everyone could kill and consume everyone. Such simulations were difficult to run. The probabilities of successfully killing another had to be kept low, at around 20%. Otherwise one of the species would quickly go extinct.

As expected, genetic distance increased rapidly and the decision maps showed a wide range of strategies under the tests conducted. The simulations were ex-

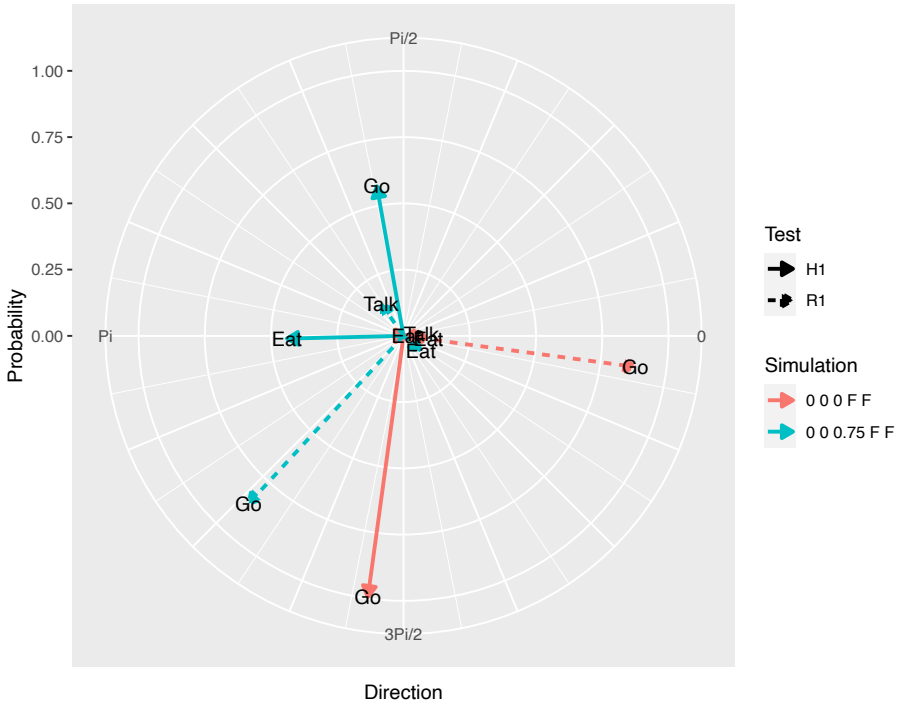

Figure 8: The graph is a decision map comparing robots evolved without selection pressure (000FF) to those evolved where humans are dangerous (0 0 0.75 F F). In both cases, the robots are tested with a human at $\alpha = 0$ (H1) and a robot at $\alpha = 0$ (R1).

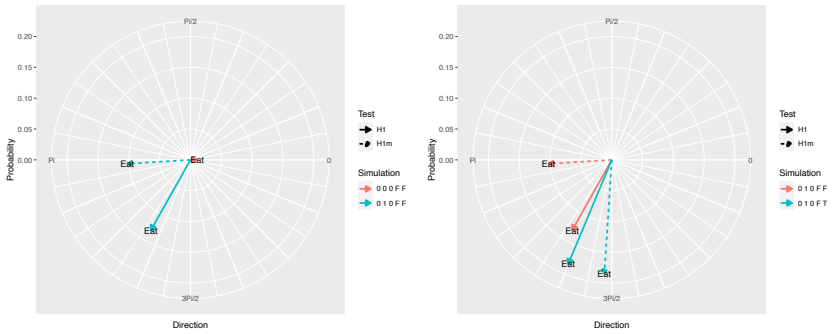

Figure 9: The graph on the left is a decision map comparing robots evolved without selection pressure (000FF) to those evolved where they have the ability to kill a human (010FF). In both cases, the robots are tested with a human at $\alpha = 0$ (H1) and $\alpha = 3\pi/2$ (H1m).

tended to 50,000 simulations and several hundred generations, but no convergence was emerging in the decision making. In this situation the robots may be evolving toward several distinct subpopulations with unique survival and replication strategies. Even in a state of warfare, the robots still return some of their resources to the depot, but this decreases with time compared to the evolutionary need to replicate.

4 Conclusions

The robots in the simulation were given brains optimized to a single goal – gather and return resources. Under selection pressure, they evolved to run from their enemies, kill their enemies, and even eat their enemies.

For evolution to occur, Charles Darwin identified these four key requirements in his famous On the Origin of Species by Natural Selection (1859):

- Variability: the encoding of a trait must vary in a population
- Heritability: the encoding must be passed to offspring
- Struggle of Existence: more offspring are created than can survive
- Probabilistic Procreation: the probability of surviving and creating offspring varies with the trait being selected

All of Darwin's requirements are present within the simulations here, and consequently the neural network brains of these AI entities are observed to evolve new behaviors in response to the selection pressures of their environment as set by the meta-parameters of the simulation.

These simulations did not evolve anything we might associate with morality. The creation of moral agents is an important research area [6], but to evolve moral agents, the simulation would need to allow better cooperation [22, 4]. The only possible cooperation here is via "talking" and sharing maps with an adjacent agent . This action does not provide sufficient mutual benefit to encourage cooperation, so these AI agents actually became less cooperative. Without the possibility of mutually beneficial cooperation, evolution encourages selfishness.

When AI entities or robots are first created, they may perhaps be carefully designed to avoid undesirable behaviors. The extent to which learning will lead an entity to undesirable behaviors is a function of how well their goals (their optimization function) are designed. Evolution is different from learning, because no matter how well designed an optimization function might be, the selection pressures of survival and procreation add a meta-optimization that will change the original intent. Worse, these selection pressures lead to competitiveness that may manifest in destructive behaviors. Evolutionary pressures will produce all the worst aspects of humanity. Until we attain the ability to embed an unalterable moral imperative, we must avoid evolving AI agents. Competitiveness is a necessary design feature of game-playing AI, network intrusion or counter-intrusion AI, and a number of other applications. In those domains, we might have some hope of control through careful design, as long as the AI entities are learning, not evolving.

Based upon these results, self-replicating AI entities are dangerous to human society. Aside from a moratorium on self-replication. Thoughtful engineers might design a fail-safe whereby an identity check must pass or else the offspring is destroyed, but even the fail-safe could develop replication errors. Various mechanisms could be envisioned to dramatically slow evolution in a self-replicating system, but designing a system with perfect replication in 100% of cases is implausible. Even to create self-replicating robots to explore or terraform other worlds risks creating the aliens that could one day destroy us. If we are to have machines replicate, then we must prevent evolution.

One solution is to remove the selection pressures that would drive evolution. This could be accomplished by providing a form of immortality. AI agents could be imbued with continuous backup so that if they are ever destroyed, accidentally or maliciously, they can simply be respawned from a backup. When there is no risk of death, as in our first simulations without selection pressures, procreation does not lead to evolution toward traits dangerous to humanity or even robot society. Evolution breeds competitiveness, and we cannot allow AI entities evolve to compete with humanity.

Source Code

The source code for this project is available at https://github.com/MLExporer/AIEvolve.

References

[1] W Brian Arthur. Inductive reasoning and bounded rationality. *The American economic review*, 84(2):406–411, 1994.

[2] Robert Axelrod and Douglas Dion. The further evolution of cooperation. *Science*, 242(4884):1385–1390, 1988.

[3] Martin T Barlow. Galactic exploration by directed self-replicating probes, and its implications for the fermi paradox. *International Journal of Astrobiology*, 12(1):63–68, 2013.

[4] Marc Bekoff. Wild justice, cooperation, and fair play: Minding manners, being nice, and feeling good. In *The origins and nature of sociality*, pages 53–80. Routledge, 2017.

[5] Rasmus Bjoerk. Exploring the galaxy using space probes. *International Journal of Astrobiology*, 6(2):89–93, 2007.

[6] José-Antonio Cervantes, Sonia López, Luis-Felipe Rodríguez, Salvador Cervantes, Francisco Cervantes, and Félix Ramos. Artificial moral agents: A survey of the current status. *Science and Engineering Ethics*, 26(2):501–532, 2020.

[7] George Church. Picogram-scale interstellar probes via bioinspired engineering. *Astrobiology*, 22(12):1452–1458, 2022.

[8] Christopher F Chyba and Kevin P Hand. Astrobiology: the study of the living universe. *Annual Review of Astronomy and Astrophysics*, 43(1):31–74, 2005.

[9] Matt Deitke, Dhruv Batra, Yonatan Bisk, Tommaso Campari, Angel X Chang, Devendra Singh Chaplot, Changan Chen, Claudia Pérez D'Arpino, Kiana Ehsani, Ali Farhadi, et al. Retrospectives on the embodied ai workshop. *arXiv preprint arXiv:2210.06849*, 2022.

[10] H Van Dyke Parunak, Robert Savit, and Rick L Riolo. Agent-based modeling vs. equation-based modeling: A case study and users guide. In *International workshop on multi-agent systems and agent-based simulation*, pages 10–25. Springer, 1998.

[11] Bruce Edmonds. Modelling bounded rationality in agent-based simulations using the evolution of mental models. In *Computational techniques for modelling learning in economics*, pages 305–332. Springer, 1999.

[12] RA Freitas. Terraforming Mars and Venus using machine self-replicating systems (srs). *J Br Interplanet Soc*, 36:139–42, 1983.

[13] Robert A Freitas and William P Gilbreath. Advanced automation for space missions. *Journal of the Astronautical Sciences*, 30(1):221, 1982.

[14] Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature communications*, 12(1):1–12, 2021.

[15] Olivia P Judson. The rise of the individual-based model in ecology. *Trends in ecology & evolution*, 9(1):9–14, 1994.

[16] Multistrategy Learning. Genetic programming: Evolutionary approaches to. *Machine Learning: A Multistrategy Approach*, page 549, 1994.

[17] Olof Leimar and Peter Hammerstein. Evolution of cooperation through indirect reciprocity. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1468):745–753, 2001.

[18] M Oprea. Agent-based modelling of multi-robot systems. In *IOP Conference Series: Materials Science and Engineering*, volume 444, page 052026. IOP Publishing, 2018.

[19] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems*, 11(3):387–434, 2005.

[20] ET Penrose. The theory of the growth of the firm (fifth impression), 1972.

[21] Lionel S Penrose. Self-reproducing machines. *Scientific American*, 200(6):105–117, 1959.

[22] Michael Tomasello and Amrisha Vaish. Origins of human cooperation and morality. *Annual review of psychology*, 64(1):231–255, 2013.

[23] Stephen Branden Van Oss and Anne-Ruxandra Carvunis. De novo gene birth. *PLoS genetics*, 15(5):e1008160, 2019.

[24] John Von Neumann, Arthur W Burks, et al. Theory of self-reproducing automata. *IEEE Transactions on Neural Networks*, 5(1):3–14, 1966.