Artificial Evolution with an Artificially Intelligent Population

Daniel Årrestad Stave
Department of Computer Science
Oslo Metropolitan University
Oslo, Norway
s351660@oslomet.no

Majdu Omar Alali
Department of Computer Science
Oslo Metropolitan University
Oslo, Norway
s346421@oslomet.no

Hanne Korneliussen
Department of Computer Science
Oslo Metropolitan University
Oslo, Norway
s361771@oslomet.no

Henrik Nøkleby Hjellup
Department of Computer Science
Oslo Metropolitan University
Oslo, Norway
s308948@oslomet.no

Negar Elimisadr
Department of computer science
Oslo Metropolitan University
Oslo, Norway
s366271@oslomet.no

Abstract: **Computational Intelligence consists of different machine learning techniques inspired by nature, including Artificial Neural Networks and Evolutionary Algorithms. When utilized in conjunction with predator-prey models, these techniques can produce highly accurate simulations of real world ecosystems. In this report, we develop a simulation model consisting of rabbits (predator) and a mixture of nutritious and toxic flowers (prey), which uses these techniques to explore how different agent-traits evolve during each iteration of the model. Our simulation shows that the rabbits are capable of adapting to their environment by improving their intelligence, which is represented by an Artificial Neural Network trained with reinforcement learning. Even though the model can be improved, for example by increasing network complexity or learning strategy, the report provides a successful predator-prey simulation which assembles properties observed in real world ecosystems.**

Keywords: **Predator-prey simulation, Agent Based Modelling, Artificial Neural Networks, Evolutionary Algorithm**

I. Introduction

There exists a large number of optimization problems which require complex system modelling to generate an optimal solution. An example of an optimization approach is the famous predator-prey model, which utilizes behaviours in, and properties of, real ecosystems. In these models, predators and prey are distributed in a predefined environment. One general goal when modelling the predator-prey system is to reach a state where the predators and prey can coexist. In this report, we use the basics of the predator-prey model to generate a model which investigates how the intelligence of predators is favoured during each iteration, or generation, of the model. This is done by assigning an artificial neural network to each predator in the model, and train it during the lifetime of the predator.

Systems addressing predator-prey interactions are widely used in science. These systems can be used to develop algorithms that can be applied in ecological, medical and industrial fields. Indeed, exploring an appropriate population model can help better understand the dynamic features of the system. Many researchers have studied population models with evolutionary dispersal perspectives, such as dispersal depending on other species. It is well understood that interaction between different species and the species' reaction to its environment is essential when developing a more realistic dispersal model for various applications. In other words, the scientific fields relevant to understanding the mechanisms of these interactions are as diverse as evolutionary biology, behavioural ecology, Eco-morphology, molecular biology, phylogeny, neurosciences, physiology, biomechanics, and robotics. The rapidly developing field of adaptive dynamics emphasizes the combined effects of evolution and ecological interactions in population dynamics.

A. *Objective*

Our proposed model utilizes a predator-prey system with evolutionary attributes in a rabbit-flower scenario. Here, the rabbits are distributed on an open field consisting of flowers with different nutrition attributes. The rabbits exhibit intelligence, which is used when deciding which flowers to eat. The evolutionary attributes of the rabbits include speed, mass and experiences from parent-rabbits. In the model we assume that food, i.e. eating of nutritious (toxic) flowers, increases (decreases) the energy of the rabbits. The rabbits lose energy by interacting with the environment, and if the energy is below or equal to zero, the rabbit dies. In addition, the relation between energy, speed and mass is described by $E = 1/2(mv^2)$, which is the equation for kinetic energy, where m is mass and v is velocity. Each rabbit in the model has its own artificial neural network, which determines how the rabbit interacts with the environment. The performance of the neural network is improved by interpreting what consequence a given decision (i.e. eating a toxic flower) has on the rabbit's health.

The object of this work is to simulate rabbits and flowers interaction, and identify which rabbit features that are favoured in evolution. Rabbits are evolved with the evolutionary features of mass and speed, and have intelligence represented by an ANN, which enables them to make decisions of which flowers to eat.

B. *Scope*

In our proposed model, we present a simulated environment of learning agents which takes inspiration from predator-prey dynamics. We also provide a detailed description of the environment for the ecosystem under consideration, and its underlying dynamics.

The report aims to show how the proposed model can perform as an intelligent system. This is done by letting the agents (rabbits) learn how it should interact with its environment with the use of reinforcement learning, in combination with evolutionary algorithms and artificial neural networks. The environment of the model consists of rabbits and flowers distributed over a two-dimensional grid. At each iteration of the model, the rabbits move across the grid. When running into a flower, it uses its intelligence to decide whether to eat it. New rabbits are generated at each iteration, and inherits features from the parent-rabbits. The training of the rabbits relies on a strategy where the agent utilizes its experience to choose which actions to make in each iteration. Included in this strategy is the use of genetic algorithms and artificial neural networks.

Our scope in this report is to utilize the described model to demonstrate how agents (rabbits) can utilize artificial neural networks to learn how to interact with an environment consisting of various nutritious and toxic flowers. Our report focuses on a simple environment, with one of the goals being to investigate how well the rabbits can adapt to their environment. Our main focus lies within developing an ecosystem where defined agent-attributes are favoured through evolution. In addition, we want to investigate which agents-attributes are considered most important by our model.

II. Litterature review

The general predator-prey model, often called Kolmogorov's predator-prey model [1], explains how the population size of different species evolve over time given a predefined growth rate. Another famous version of the model is that proposed by Lotka and Volterra in 1926, consisting of differential equations used to explain population evolution in a simple ecosystem [2]. The solutions of these equations are periodic oscillations, meaning that it repeats itself over time. This model, and variations of it, have been proven useful in many domains. Some examples are modelling of epidemics [3], ecology [2] and economics [4]. Even though the Lotka-Volterra equations are applicable in many situations, it has been argued that these equations aren't sufficient for modelling natural phenomena. For instance, the equations do not take into account important attributes like the intelligent and adaptive behaviour of agents (predator or prey) in complex environments. As a consequence, there have been developed various types of models that exploit different machine learning techniques in order to produce more accurate simulations.

[6] is an example of a study that uses the basics of a predator-prey model together with advanced techniques in order to make the simulations of ecosystems more naturally accurate. In the study, each agent in a predator-prey environment was given their own neural network as a "brain". The neural network in [6] was used by the agents to learn the best action to take in given situations with the use of a train and error process known as reinforcement learning, which is a branch of machine learning algorithms that make it possible for agents to learn and improve their behaviours based on interactions with their environment. This is done by providing rewards for given actions performed by the agents, which then use this reward to develop the optimal policy. This policy is used to determine which action the agent should take in a given state. The presented model consisted of predator-, prey- and grass-agents that made decisions about their next move based on their perception of the neighbourhood and their learned weights. Some important aspects of this model compared to regular predator-prey models are that agents do not randomly move around in their environment. Instead, they are making decisions learned from past experiences. A similar study was performed in [5]. Here, the authors also utilized deep reinforcement learning algorithms in order to make the agents in their model capable to learn. The model used a Monte Carlo simulation algorithm to simulate the evolution process. The combination of these two algorithms made it possible for the authors to see how the simulated agents use experiences to make decisions determining their actions in the environment, as well as pass on experiences to their offspring.

The study presented in [7] modelled a three-species predator-prey ecosystem, similar to [6], where the agents were trained through multi-agent reinforcement learning. The population dynamics were then analysed together with the standard Lotka-Volterra equations. The authors developed a happiness network used as a reward system in the reinforcement learning procedure. In the study, the main goal of the agents was to collect as many rewards from performed actions as possible. This was done by exploiting learned knowledge about actions it had discovered to be good, but also exploring unknown actions.

Many of the approaches presented thus far use some form of reinforcement learning together with a predator-prey model and unsupervised learning. Some advantages of these techniques is that they make it possible to see how agents evolve their intelligence, and thereby choose smart actions, based on their perception of the environment. This means that there is no need to introduce training data where given states are mapped to given actions. Some challenges with the presented studies is that they focus on complex learning techniques and networks and do not take into account how various environmental factors may change the behaviours and learning capacities of the agents.

Taking a different approach, the study in [8] presents an individual-based predator-prey model where the behaviour of each agent was modelled by a fuzzy cognitive map. This map enabled the agents to evaluate its environment and internal state, and use this to choose the best available action. The advantages of individual-based modelling, as opposed to multi-agent modelling used in [7], is that it allows the consideration and evolving of "intelligence" of individual agents, instead of considering the entire ecosystem as a whole. Even though fuzzy cognitive maps present an interesting way of model agent behaviour it does present some challenges, one of them being the need of experts to develop fuzzy sets and membership functions, instead of using the existing, well described models of predator-prey simulations.

[10] combined genetic algorithms and a predator-prey model to find the solution to a specific optimization problem. This was done by letting the prey represent solutions to the problem and predators killing the weakest prey, which were the ones that represented the weakest solution. Genetic algorithms were also used in [9], presenting a set of artificial agents that lived in an artificial world with hazards and food. Each agent had their own neural network brain that controlled their movement in different situations. Here, a chromosome encoded the network structure and weights. This chromosome was used to evolve the system with the use of a genetic algorithm, so that profitable behaviours were retained in later generations. Each agent was given a numerical amount of energy that decreased when the agent consumed food and decreased when the agent performed various actions, like moving or mating. If the energy reached zero, the agent died. The energy value of the agents was used as a fitness score. One advantage of using a genetic algorithm like these studies have done is that it provides control over the evolution of the population and makes it possible to define which attributes that should be favoured. It also provides the opportunity to see how various attributes are favoured through evolution.

As stated in the introduction, our goal for this report is to develop a predator-prey model that simulates natural phenomena, by using reinforcement learning and evolutionary algorithms in a predator-prey simulation. More specifically, we want to see how agents in a simulation learn to coexist by interacting with other agents and the environment, similar to what was done in [5,6,7]. In addition, we want to combine this model with an evolutionary algorithm to see how intelligence is favoured during evolution as proposed in [9,10], where parent agents transfer their neural network (learned knowledge) to new agents (offspring). With regards to other research methods, we want to use some of their ways of evaluating the performance of our predator-prey simulation, like investigating if our model exhibits similar dynamics to those described by the LV equations. The intelligence can be evaluated by comparing the survival capabilities of the trained/evolved agents in our model with regular agents with random policies.

III. Theory

In our report, we use the basics of the famous predator-prey model in order to simulate complex behaviour in a nature-like ecosystem. This kind of model has been an important factor of mathematical ecology, which studies how populations of different species interact with each other. In the predator-prey model, there are two species; the predators and the prey. Examples of such real-world systems in nature are lions and gazelles or birds and insects. The main goal of the predators in such models is to catch and eat the prey, while the prey's main goal is to survive. The purpose of these models is to see how different kinds of attributes, like population growth, affects the ecosystem.

Computational Intelligence is a broad term that refers to computers ability to learn a specific task from data. Generally speaking, Computational Intelligence is a set of nature-inspired methods and technologies that aims to model and solve complex real-world problems. This report utilizes three main techniques within this field, namely evolutionary algorithm, artificial neural network, and agent based modelling (ABM). In ABM a system is modelled as a collection of agents, where each agent interacts with its environment and makes a decision in a given situation based on a set of rules. These rules can be explicitly defined (i.e. when approaching another agent, kill it) or implicitly defined (use experience of previous actions to determine the best decision to choose in a given situation).

A. Evolutionary Algorithm

Evolutionary Algorithms (EAs) present a method of solving optimization problems which take inspiration from biological evolution. The problem is solved by generating, evaluating and modifying a population of possible solutions.

The first part of the algorithm involves generating a population of solutions to a given optimization problem. This population can, for example, be a set of weights in a neural network or definitions of how various agent attributes should be favoured in a defined environment. It is important that the population includes a large number of different solutions, relevant to the optimization problem. Once the population is generated, the algorithm evaluates the performance of each solution with the use of a predefined fitness-function. The fittest solutions are then chosen to become parents. The attributes of the chosen parents get combined in a process called crossover, where the results are new solutions, named offspring, which contains attributes from the parents. In order to increase diversity in the population, mutation is included in the algorithm, making changes to the offsprings attributes. Selection, crossover and mutation ensures balance between exploration of new solutions and exploitation of generated solutions by the algorithm. This evolutionary process of selection, crossover and, possibly, mutation is repeated until a termination criterion is met. This criterion can be a defined maximum runtime of the algorithm, or a reached threshold of performance. Once the algorithm has terminated, a final solution is selected and returned, as displayed in Figure 1 [11].

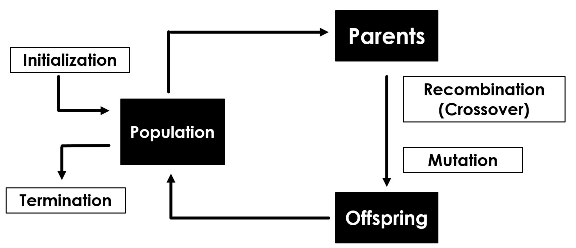

Figure 1: An overview of an evolutionary algorithm, which identifies the fittest solution in a population based on a given optimization problem, and returns the result

A wide variation of evolutionary algorithms exists, where Genetic Algorithms (GAs) is a popular choice. Here, the proposed solutions in the population are referred to as genomes, which represent a collection of variables describing the attributes of the solution. These variables, or genes, typically have a direct mapping to the problem being solved, where each represents a value that the algorithm aims to optimize. In these cases, mutation is normally implemented by replacing some of the genes in the chromosome with a new random sample from the population.

EAs presents a methodological framework which is easy to utilize in a wide range of optimization problems. As an example, they can be used in dynamic systems where the optimization goal varies according to changes in the environment. An important aspect when working with EA is the design of the fitness functions, that controls which solution is favoured within each runtime (generation) of the algorithm.

B. Arificial Neural Network

Artificial Neural Networks (ANNs) can be described as a computer system consisting of highly interconnected processing nodes. These nodes are organized into layers, where the nodes of each layer are connected with notes in their neighbouring layers, as displayed in Figure 2. These layers include one input layer, which receives input from a problem domain, hidden layers, which produce patterns to define the mapping between inputs and outputs, and one output layer. Generally speaking, ANNs aim to solve an optimization problem by receiving inputs from the problem domain and generating an output, with the use of network parameters. One of these parameters is the weight of the connections between each node. These weight-values determine how an output is calculated from a given input, and can be modified based on an evaluation of the produced outputs. This implies that a good output causes small weight-changes, while an output which is considered less optimal causes a bigger modification of the network weights.

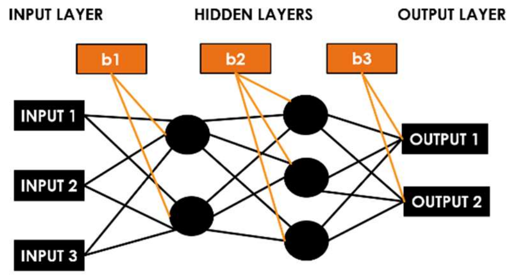

Figure 2: A simple visualization of an ANN, consisting of nodes organized into layers. The weights of the connections among the nodes can be altered to produce optimal outputs for given inputs

Each node in the network is equipped with an activation function, which determines how the node should interpret received output. The activation function utilizes the summed weight of each connection to the node, and produces an output-value which is passed forward in the network. Common activation functions include the sigmoid function and the rectified linear unit function. There exist various methods for altering the network weights, a common choice being backpropagation with gradient descent. This technique uses a loss function, which evaluates the produced output. The aim when altering the weights then becomes to minimise this loss function.

It is normal to divide the learning techniques used in the fields of computational intelligence into supervised, unsupervised and reinforcement learning. In supervised learning, the system is provided with a set of inputs and optimal outputs, which is used to generate a pattern that maps these two together. When presented with new inputs, where the optimal outputs are unknown, the system can use the generated pattern to produce optimal outputs. In unsupervised learning, these initial input-output sets are not presented to the system. In these situations, the system works by looking for underlying patterns and features in the input-data, which then is used to generate output data. In reinforcement learning, the system learns by interacting with its environment. Here, the system receives positive feedback when producing optimal outputs which has a positive impact on the environment and negative feedback when the outputs are less optimal, meaning they have a negative impact on the environment.

The advantage of utilizing ANNs to solve an optimization problem includes its ability to process incomplete data and recognize patterns. It is also adaptive, meaning it is able to adapt to changes in the problem domain. Some challenges include the fact that most neural networks are dependent on a training phase, where they learn optimal weights. Therefore, the quality of the data greatly depends on the quality of data utilized in the training phase. Reinforcement learning presents one way of overcoming this challenge, by letting the network use feedback from its environment as training data. Even though ANNs generally perform better with more hidden layers and units, it is important to consider the effect this has on required computer power and learning time. Another problem that may arise is the one known as overfitting, which makes the model unable to generalize well to new input data.

C. *Artificial Evolution*

Artificial Evolution can be described as a technology which combines aspects from the fields of artificial intelligence and natural evolution. This combination makes it possible to evaluate and evolve the artificial intelligence system with use of concepts inspired from evolution in nature, like the use of fitness and selection to determine which solutions should be passed to the next iteration, or generation, of the system.

Our research problem consists of studying and exploring the capabilities of artificial evolution with an artificially intelligent population. We define artificial evolution as any algorithm or method using the mechanisms of Darwinian evolution to generate a product in an artificial, or non-biological manner, whereas intelligent population is defined as a population where each individual is granted a type of artificial, non-biological intelligence, such as an artificial neural network. Together they can be combined into a complex model.

IV. Methodology

In this section, we begin with presenting the general outline of our simulation, followed by a detailed explanation of the different agent-types in the model, namely the flowers (prey) and the rabbits (predator). Then, we describe the techniques used to train and evolve the agents, specifically ANNs and EA.

Our model simulates the interactions between rabbits and flowers. Rabbits jump randomly around in a 100x100 grid with periodic boundary conditions. If they encounter a flower, they decide whether or not to eat it with the use of their intelligence represented by ANNs. Flowers spawn at random locations. The environment used in our model can be described as a field where rabbits can move freely and eat available flowers. This environment consists of three parts: flowers, rabbits, and vacant ground. There exist five different types of flowers, where some of them are toxic and can decrease the health of the rabbits when eaten, whereas some are nutritious and increase the health of the rabbits. At the beginning of the model simulation, the rabbits eat random flowers, not taking into account how nutritious (or toxic) they are. As the simulation continues, the rabbits develop an experience of which flowers they should eat or not in order to increase their health points. One important feature of the rabbits is their speed. A rabbit with higher speed can move more than one grid per iteration, but at the same time utilize more of its energy. In other words, rabbits will lose more health points by moving faster in the field. An important thing to mention is that the rabbits have mating ability. This feature makes it possible to include different generations in the simulation and to transfer experience and features from one rabbit to another.

A. *Flowers*

The flowers represent the prey in the predator-prey system, and are one of two agent-types that exists in our model. Some flowers are nutritious, others toxic. If a rabbit decides to eat a flower, it is affected in two ways

- The health of the rabbit is updated, positively if the flowers is nutritious, negative if the flower is toxic
- The ANN weights of the rabbit is updated through backpropagation, correct classification if flower is nutritious, incorrect classification if the flower is toxic

We have 5 different flowers, each with a unique nutrition profile with respect to their size, as displayed in Figure 3. In the simulation, these values are scaled by a factor of 5. The flower size is treated as an integer number and their size are numbered 1 to 5, with increasing variation in nutrition, and thus degree of difficulty with respect to classification. The introducing of flowers with non-linear decision boundary requires that the ANN has non-linear activation functions. The functions used to generate the flowers are:

- Flower 1, $nutrition = 1$
- Flower 2, $nutrition = -1$
- Flower 3, $nutrition = \frac{size-1}{9*2-1}$
- Flower 4, $nutrition = \frac{1-size}{9*2+1}$
- Flower 5, $nutrition = \cos(size)$

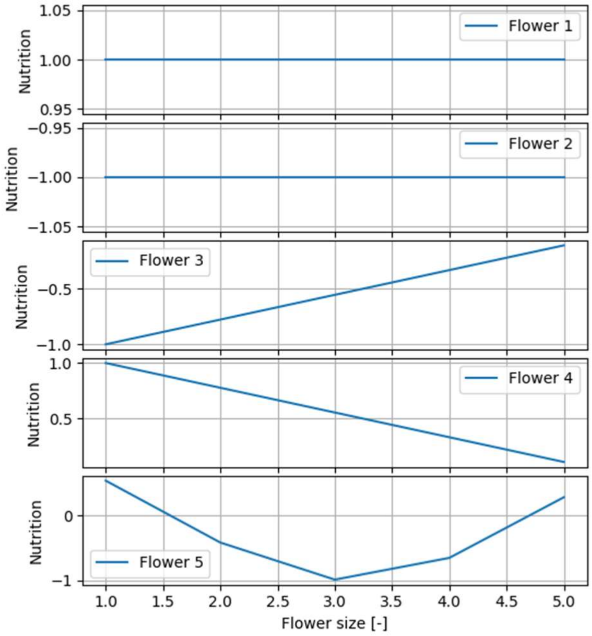

Figure 3: The nutrition valve with respect to flower size for the 5 different flowers

B. Rabbits

The rabbits represent the predator in the predator-prey system, and have three features: speed, mass and intelligence. The speed lets them cover more ground, i.e., search more space for flowers. In this work, the rabbit's speed is the number of grid cells it searches for food per simulation iteration. The mass lets them digest the flowers they eat better. The digestion efficiency, d_{eff}, is defined as

$$d_{eff} = 100\left(\frac{1}{2(1+e^{7-2m})} + \frac{1}{2}\right) \qquad (1)$$

with its digestion effect visualised in Figure 4. A digestion efficiency of 50 % returns 50 % of the nutrition, regardless of positive or negative nutrition value. The m in Equation 1 is the rabbit's mass. Intelligence makes it possible for the rabbit to make better flower eating decisions, and is defined as the accuracy of rabbits ANN for predicting positive or negative nutrition value for all flowers at all sizes. Since we have binary classification, the default accuracy with random guessing is 50 %.

Each rabbit has a float-valued health. At each simulation run, the rabbit's health is subtracted by energy consumption dependent on its features

$$\triangle H = \frac{\frac{1}{2}mv^2 i}{100} \qquad (2)$$

where $\triangle H$ is the energy consumption, m is the mass, v is the velocity and i is its intelligence. Rabbits of the initial population are initialised with a health of 500. If the rabbit's health reaches 0, it is considered dead and removed from the simulation. New rabbits are initialised with health equal to average rabbit health of the current (living) population. The rabbits are evolutionarily evolved, and Equation 2 forces the rabbits to choose a set of features that maximises rabbit health and minimises energy consumption. The rabbits' mass and speed are treated as integers.

We investigate how the rabbits speed, mass and intelligence is favoured in the simulation with cyclical changes in the environment. The set of features that maximises the population size are of particular interest. We hypothesize that the best individuals have high intelligence combined with either high velocity or high mass. From analysing digestion efficiency improvement with respect to mass in Figure 4, we see that masses above 6 kg causes a negligible improvement. Yet, higher masses are punished in Equation 2. Thus, we expect the evolution of the mass not to exceed 6 kg. The lower limit of speed and mass is set to 1. Upper limit of mass is 10, with no upper limit for speed. The initial rabbits, i.e., those who isn't a result of the evolutionary algorithm, are initialized as follows

- mass, 4 plus a random number from a 0 mean 2 standard deviation normal distribution
- velocity, 8 plus a random number from a 0 mean 2 standard deviation normal distribution
- ANN weights, a random number from a 0 mean 0.1 standard deviation normal distribution

The initial rabbit population size is 100, and the initial flowers size are subject to randomness with an approximate initial population of 500. For the first 50 iterations, the minimum flower population is set to 300. After that, the minimum flower population, fp_{min}, becomes a periodic function of iteration, t, to challenge the rabbit evolution, according to the equation

$$fp_{min} = 200 + 100\sin\left(\frac{t}{3}\right) \qquad (3)$$

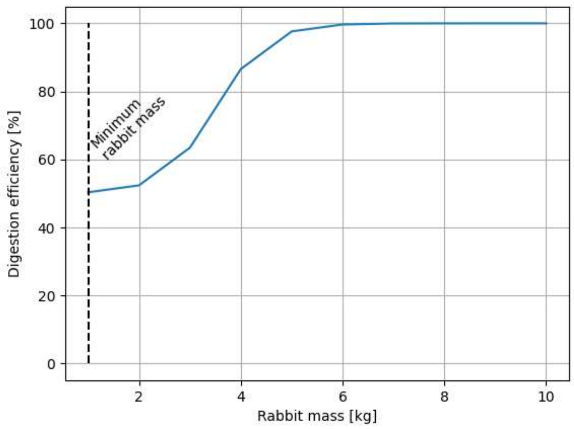

Figure 4: The rabbit digestion efficiency with respect to rabbit mass

C. *Evolutionary Algorithm*

Rabbits are bred evolutionary with the use of a genetic algorithm, where the fitness value is their health. The parents are selected through tournament selection with 3 randomly chosen opponents. Crossover probability is set to 50 %, otherwise children are copies of their parents, and mutation of each index in the genotype is 50 %. Their genotype can be considered a list of speed, mass and flower eating experiences, where the latter has no mutation. Speed and mass is mutated by adding a number of a 0 mean 2 standard deviation normal distribution. Offsprings do not replace individuals in the population. Four rabbits are born every simulation iteration.

D. *Arificial Neural Network*

The structure of the rabbit's ANN is 6 input, 1 hidden layer with 4 nodes, and 1 node in output layer. The nodes in the hidden layers have rectified linear unit activation. The output layer has sigmoid activation function. The output is rounded to the nearest of either 0 or 1. 0 is a decision not to eat a flower, and vice versa. 5 of the ANN inputs are used for colour encoding of binary type, and the 6th for flower size. The colour encoding can be described by the following example. If a rabbit encounters a flower of size 2, and blue colour, the flower is encoded as follows

[0,1,0,0,0,2]

Or for a black colour with 4 size

[1,0,0,0,0,4].

For the size of the flower, we apply feature normalisation to have the size in range -0.5 to 0.5. This is done by Equation 4.

$$flowersize^{*} = \frac{5 - flowersize}{4} - \frac{1}{2} \qquad (4)$$

A decision of eating a flower with positive nutrition value, as shown in Figure 3, is considered a correct classification. Eating a flower with negative nutritional value is considered an incorrect action, and the weights are adjusted accordingly in the backpropagation with gradient descent.

The rabbits store the decision and consequences in their memory, which is used when updating their ANN. The update is done after every new decision, with a learning rate of 0.01 and 1000 iterations, and when a new rabbit is evolved with the use of their parents' memories. The length of their memory is restricted to 1000 experiences, to lower computational expense. If this limit is reached, the newest experience will replace the oldest experience.

V. RESULTS AND DISCUSSION

In this section, we provide an overview of our simulation results together with a discussion. We start by presenting and describing the results before providing argumentation for our chosen parameters. Then, we compare our model to other versions of the predator-prey simulation and ecosystems in the field of biology. The purpose of this comparison is to provide argumentation for why our model can be used to describe natural phenomena.

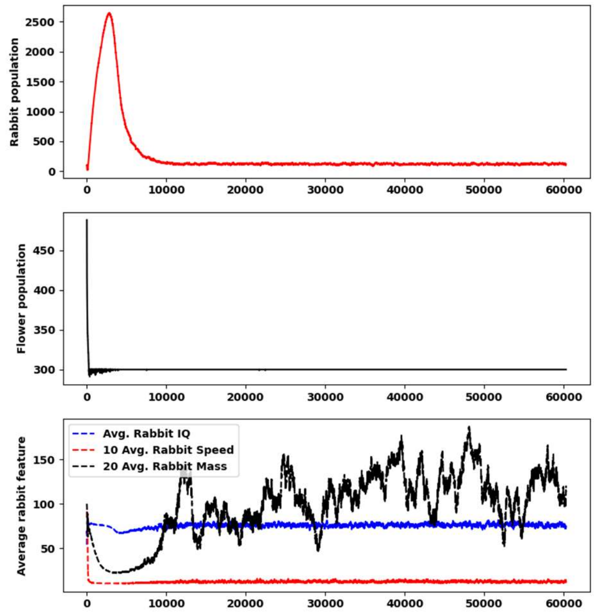

Figure 5: Simulation results of our model over 60000 simulation iterations. As illustrated, the model stabilizes to a population of around 120 rabbits

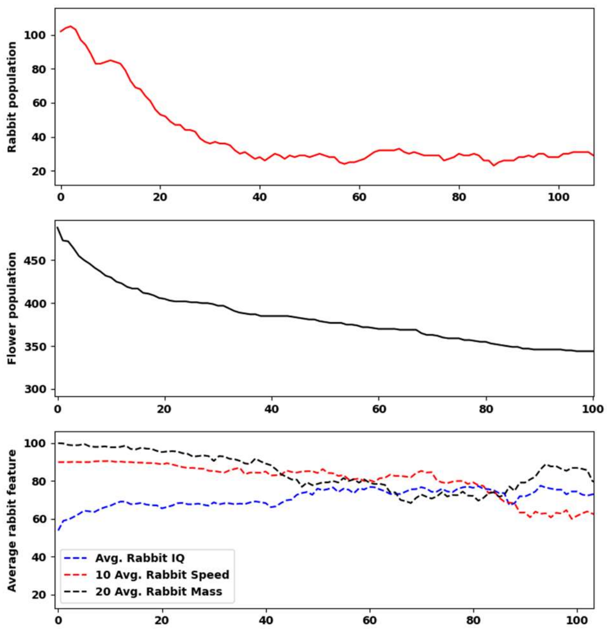

Figure 6: The initial iterations of the simulation shows how the population of rabbits and flowers, and attributes of the rabbits, evolves

Figure 5 shows the simulation results for the first 60 000 iterations. Figure 6 shows a zoomed image of the first 100 iterations.

The rabbit population stabilizes after around 10 000 iterations. The rabbit population increases to a peak value at 2646, before declining and stabilizing to a population of approximately 120 rabbits after 10 000 iterations. As for the flower population, which is controlled by Equation 3, we that it reaches an average value of 300. Given the fact that flowers are set to have a cyclic minimum population of 100-300 flowers, this indicates that the rabbits are unable to eat all available flowers in the environment when the minimum flower population are at 100, before it rises to 300. This is mainly because of the small rabbit population. The rabbit intelligence/IQ starts at 50 % and rises to approximately 80 %. It has a slight dip at iteration 5000, but eventually reaches a stable value of 70-80 %. The rabbit speed quickly decline to the minimum value of 1 and remains at this value for the remaining simulation iterations. This indicates that the speed was to heavily punished in Equation 2. The rabbit mass, similar to speed, seems to drop rather quickly in the initial iterations. After about 5000 iterations the trend seem to turn and larger rabbit-mass are favoured. Considering the digestion efficiency visualised in Figure 4, we see the digestion efficiency for mass in the range of 2 - 6 kg is recognized in the evolution. Keep in mind that the graph for rabbit mass is multiply by 20. This means that horizontal y-axes values are 40-120 for 2-6 kg. For rabbit masses above 6 kg, there are no additional benefits in digestion efficiency, which the evolution seems to figure out. The rabbit mass fluctuates between 3-7 kg, which corresponds to 60-140 in Figure 4.

Intelligent rabbits with moderate mass were favoured in the evolution. The evolution took partly advantage of the digestion efficiency associated with larger rabbit masses, which is displayed in Figure 4. A high mutation probability enabled vast searches for mass and speed, still, only the mass seems to be evolutionary explored. A low number of opponents in the tournament selection compared to the population size, also principally allowed for weaker individuals to breed. Overall, the evolutionary parameter selection is considered successful.

The described ANN structure, with a learning rate of 0.01 and a fixed 1000 iteration update, enabled the rabbits to learn the binary classification of healthy vs toxic flowers rather quickly, as shown in Figure 6, with intelligence rising quickly. After 50 iterations, the population average intelligence is 75 %. After 10 000 iterations, it is at 70-80 %. Thus, we deem the ANN structure, parameter and hyperparameter selection partly successful. The network structure with the same hyperparameters were trained on a full training set and reached an accuracy on if 100 % after 50 000 iterations. Thus, to further improve the model, it is advised to train the ANN for more iterations and perhaps start each update with a fresh set of random initialised weights, instead of continuing with the previous optimised weights.

With the scope of our report being the investigation of how various agent attributes are favoured during evolution, we utilized several model assumptions which can be altered in order to provide more realistic results from a biological perspective. One of these assumptions is the selection of parents in the evolutionary process. Our model uses tournament selection, where the fittest parents from the entire population are included as parent-candidates. One more realistic approach would be to only allow agents close to each other to be considered as mating partners in the selection process. In addition, the agents in our model do not lose any health points when mating, i.e. generation of new offspring, is performed. Another important aspect of our model is the fact that the rabbit population does not have an upper limit. Instead, the rabbits learn to adjust their attributes to avoid overpopulation.

A. Results compared to other work

Our model represents a version of the predator-prey model where the rabbits represent the predator and the flowers represent the prey. As mentioned in the literature review, the Lotka-Volterra equations represent predator-prey dynamics in their simplest form. These equations make several assumptions, including no environmental complexity or population limitations. However, the equation is useful for visualizing expected general properties of a predator-prey model. The LV model predicts a cyclical relationship between the population of predator and prey, meaning that an increase of the number of predators results in a decrease of the number of prey.

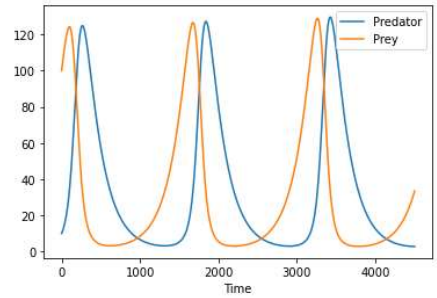

Figure 7: Visualisation of the LV equation, showing that the predator and prey population affects each other and that, in a successful simulation, oscillations form which ensures survival of both predators and prey

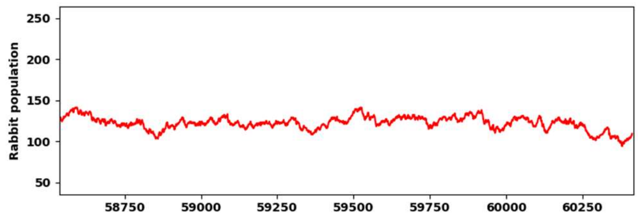

Figure 8: The rabbit population stabilises in the later iterations to a size of approximately 100-140 rabbits

Although the LV model is quite different from our model, both in definition and simulation, we can see similar behaviours when zooming in to the time in our model where the rabbit-population has stabilized. This leads to the assumption that the rabbit population is dependent on the amount of available flowers. When comparing the beginning and end of our simulation, we see that the rabbits use a considerable amount of generations to reach this state. An argument for this is that it is expected that an initial population with no evolved attributes will require time to adjust to its environment. But once it has, the rabbits (predator) and flower (prey) seem to coexist in a constant manner, not that different from the properties explained by the LV-equations.

The main feature of our model is the inclusion of artificial neural networks, which is used by the rabbits when making decisions about which flowers to eat. One of our goals is this report is to prove that intelligence, in the form of identifying nutritious and toxic flowers, is an important aspect of this system. To demonstrate the importance of this attribute, Figure 9 shows a simulation where the rabbits make their decision at random, as is the situation in many early versions of the predator-prey system. As the figure shows, the population of rabbits stabilizes at a much higher value. This is because by Equation 2, lower intelligence requires less energy. Thus, the given flower population can carry more rabbits. Further, the evolutionary features of the rabbits are all converged to its minimum value, 1 for mass and speed and 50 % for intelligence. In this simulation, the rabbits are able to eat all of the available flowers because of its large size. This makes the cyclic attributes of the flower population more obvious.

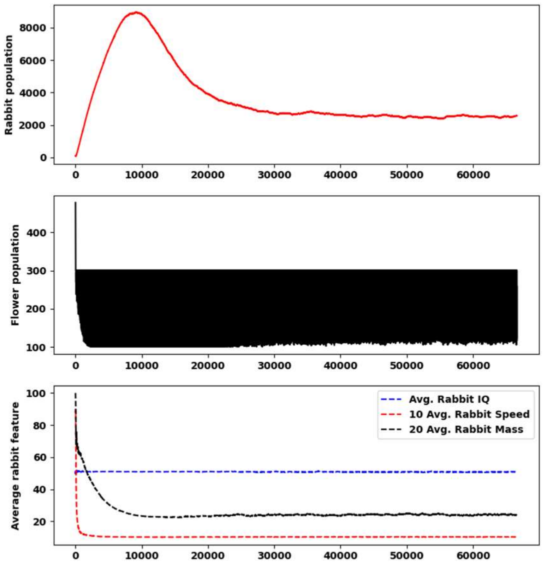

Figure 9: Shows simulation results when rabbits perform random actions

B. *Results compared to biology*

As previously stated, the predator-prey model is commonly used when investigating the dynamics of real world ecosystems. Characteristics that are often favoured in these models include the ability of the predators and prey to coexist, by introducing parameters and rules that describe how each agent in the model should interact with other agents and the environment. Since the main goal of these models is to simulate real world ecosystems and show how various features of the agents are favoured, a good method of model evaluation is to compare its results with analysis and research conducted on real ecosystems.

One popular area of research within biology is Darwin's theory of evolution, which, generally speaking, states that individuals with favoured traits within a population will allow them to survive by being able to adapt to their environment. The traits of these individuals will, after generations, become more frequent in the population leading to an evolution of the individual's traits. When analysing the rabbit's traits in our simulation model, we see that intelligence is favoured throughout evolution. This seems like a reasonable result, since the consequence of a low intelligence increases the risk for the rabbit to eat potential toxic flowers. When looking at the evolution of traits in the population, we can see that the rabbit seems to test out various mass and speed attributes, before settling at a constant minimum low speed value and constant moderate to a high mass value, as shown in Figure 10 and 11.

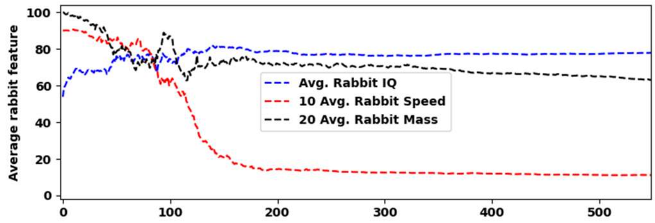

Figure 10: Shows how the various traits are favoured in the initial part of the simulation

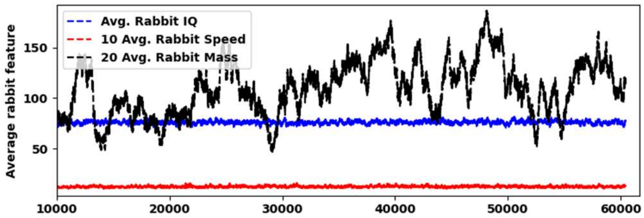

Figure 11: Shows how the various traits are favoured in the later part of the simulation

This evolution of agent traits is a result of the ability of the agents to transfer its experience, i.e. ANN, to its offspring. The use of tournament selection when deciding which agent should be parents ensures that only those that perform well in the environment pass on their attributes, including speed, mass and experience. One important observation that can be made from our model is that the rabbit's intelligence (IQ) quickly converges to a steady high value, while the properties of the flowers remain the same. A more realistic scenario would be to introduce more challenges to the environment, like obstacles, additional flowers or predators, that forces the rabbits to utilize its learnt intelligence at a wider range.

VI. CONCLUSION AND RECOMMENDATION

In this work, we successfully developed a predator-prey simulation with flowers and rabbits. The rabbits learned which flowers to eat and at what size using their own ANN. The ANN was trained using their own and their parents' experiences as training examples. Rabbit intelligence quickly rose to around 80 %, and remained high for the remaining simulation. Mass of the rabbit, as a representation of digestion efficiency, proved to be the second most important rabbit feature. The rabbit's speed, i.e., how many grid cells it searched for food during each iteration, was the least evolutionarily favourable feature. The rabbit's features successfully adapted to the environment through evolution. The rabbits prioritise intelligence over speed and mass, but eventually converged to intelligent rabbits with a moderate to maximum value of mass and a minimum value of speed.

The simulation model exhibits similar properties as demonstrated in [6], where agents in a predator-prey system utilized past experience to train a neural network, which was used to conclude optimal decisions for given environmental states. In addition, our model shows basic ecological properties, like a population stabilization dependent on environmental resources (flowers) and favouritism of specific agent traits, which enables the agent to perform well in, and adapt to, their environment.

There are multiple avenues of exploration to continue the work described in this report. The Authors have the following recommendations for expansion of the work, and to researchers or practitioners doing similar work

- Investigate other rabbit intelligence representations for AI. The ANN could potentially be replaced by spike ANN to reduce computational power
- Modify the delta health function (equation 2) to have less punishment for speed and more punishment for intelligence to obtain more balanced features of rabbits
- Add more complexity to the environment, like obstacles, or other predators, e.g., foxes
- Add vision to rabbits
- Make rabbits able to learn from attributes they observe in the environment, e.g., if a rabbit eats a toxic flower the rabbits nearby also learn from the experience
- Make the flowers or other environmental factors harder to learn, to give development of intelligence a greater challenge
- Apply neuroevolution to rabbit ANN, possibly by the method known as NeuroEvolution of Augmenting Topologies (NEAT)
- When updating the rabbit ANN after the first time, it may be better to start with random weights and perform more iterations rather than to continue with the previous weights

References

[1] F. Hoppensteadt, "Predator-prey model," Scholarpedia, 2006.

[2] [E. W. Kopf and A. J. Lotka, "Elements of Physical Biology," Journal of the American Statistical Association, vol. 20, no. 151, 1925, doi: 10.2307/2965538.

[3] S. Bhattacharya, M. Martcheva, and X. Z. Li, "A predator-prey-disease model with immune response in infected prey," Journal of Mathematical Analysis and Applications, vol. 411, no. 1, 2014, doi: 10.1016/j.jmaa.2013.09.031.

[4] D. W. MacKenzie, " The Predatory State by James Kenneth Galbraith ," American Journal of Economics and Sociology, vol. 69, no. 2, 2010, doi: 10.1111/j.1536-7150.2010.00718.x.

[5] X. Wang, J. Cheng, and L. Wang, "Deep-reinforcement learning-based co-evolution in a predator-prey system," Entropy, vol. 21, no. 8, 2019, doi: 10.3390/e21080773.

[6] Andrea Ramazzina, "Reinforcement Learning in a Predator Prey model," 2018.

[7] Tobias Karlsson, "Multi-Agent Deep Reinforcement Learning in a Three-Species Predator-Prey Ecosystem," 2021.

[8] P. Sunehag et al., "Reinforcement learning agents acquire flocking and symbiotic behaviour in simulated ecosystems," 2020. doi: 10.1162/isal_a_00148.xml.

[9] C. R. Ward, F. Gobet, and G. Kendall, "Evolving Collective Behavior in an Artificial Ecology," Artificial Life, vol. 7, no. 2, Apr. 2001, doi: 10.1162/106454601753139005.

[10] X. Li and S. Sutherland, "A cellular genetic algorithm simulating predator-prey interactions," in Proc. 4th Asia-Pacific Conference on Simulated …, 2002.

[11] Devin Soni, "Introduction to Evolutionary Algorithms", https://towardsdatascience.com/introduction-to-evolutionary-algorithms-a8594b484ac