EG UK Computer Graphics & Visual Computing (2020), pp. 1–5
P. D. Ritsos and K. Xu (Editors)

# Simulating Dynamic Ecosystems with Co-Evolutionary Agents

paper1015

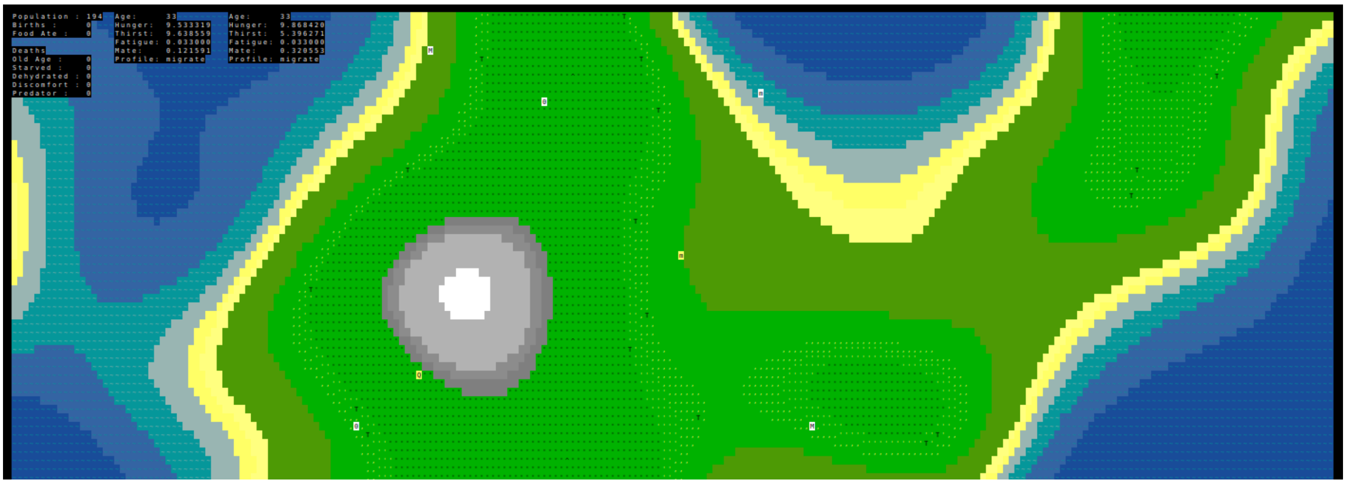


**Figure 1:** *Visual output of the program, showing the environment populated with a variety of agents.*

**Abstract**

*As video games grow in complexity and require increasingly large and immersive environments, there is a need for more believable and dynamic characters not controlled by the player, known as non-player character (NPC). Video game developers will often face the challenge of designing these NPCs in a time efficient manner. We propose an agent-based Cooperative Co-evolution Algorithm (CCEA) where NPCs are implemented as artificial life (AL) agents that are created through an evolutionary process based on simple rules. The virtual environment can be filled with a range of interesting agents, each acting independently from one another, to fulfil their own wants and needs. We proved that agents implemented with a very limited number of variables making up their genome can be successfully integrated in a co-evolutionary multi-agent system (CoEMAS). Results showed promising levels of speciation and interesting emergent and plausible behaviours amongst the agents.*

**CCS Concepts**

• ***Computing methodologies*** → ***Image processing; Artificial life; Agent / discrete models;*** *Real-time simulation;* • ***Applied computing*** → ***Computer games;*** • ***Mathematics of computing*** → *Evolutionary algorithms;*

## 1. Introduction

This short paper explores the usage of co-evolutionary multi-agent systems (CoEMAS) for simulating dynamic ecosystems filled with artificial life (AL) agents capable of generational evolution. CoEMAS are a form of evolutionary algorithm (EA) that takes a decentralised approach to selection in breeding and divides a problem across several sub-populations, wherein the sum of these parts equates to the solution. The solution of the problem being the ecosystem with each sub-population fulfilling a niche within it.

Using CoEMAS to create AL agents will allow for an ecosystem that displays a wide range of speciation (i.e. formation of new and distinct species during evolutionary loop) and emergent behaviours. It will also give an easy to understand look into the machinations involved in Darwin's theory of evolution as we see the rise and fall of species. The agents in this system will be able to exhibit a range of behaviours based on their genetics. For example, in our simulated environment, there are three main species: herbivores, scavengers, and predators.

This project aims to provide a framework that could be used as a component within a video game. This system would provide players with an immersive world filled with dynamic and rich wildlife. As wildlife can change generationally, as time passes the fauna is less likely to become predictable. This would mainly come in use for games that rely heavily on immersive worlds filled with a variety of fauna. For example, you might use this system as the basis of a game wherein the player is set with the task of studying the ecology, possibly even being provided with tools that could cause changes to the environment and thus forcing evolutionary changes in the wildlife.

As well as its application within the video game development industry, this project could also be used in a research context. As the framework is loosely based off concepts displayed in real-world evolution, it may also be used to replicate that, allowing for a user to better understand how particular nuances of evolution might take place.

## 2. Related Works

### 2.1. Co-evolutionary Multi-Agent Systems

The foundations for this project are based on co-evolutionary algorithmss (CEAs), a form of EAs that implements the process of co-evolution, wherein a change in one species will result in a response from the others. In a CEA each species may be trying to solve a separate problem from each other or taking a different approach to optimising a problem [JWSL13]. The benefit of this approach is best seen in problems that might not have a tangible way of formulating an explicit fitness function or require more population diversity.

This can be further expanded upon with CoEMAS. This system allows the problem to be divided across several sub-populations, with the whole population providing the solution. Agents within this system make use of three profiles: resource, reproduction, and migration. In the resource profile the agent can seek resources, get them, or die. The reproduction profile allows them to seek out a partner, clone themselves, perform recombination of the two individuals' genetics, as well as mutations. The agent is also then able to give some of its resources to the new offspring [DS08].

### 2.2. Artificial Life in Entertainment

AL is a wide field that covers a wide array of subject matters, from artificial ecosystems, evolution, to morphogenesis, and much more. In the words of Chris Langton, founder of the field, the literal and yet purposefully vague definition of AL is "life made by humans rather than by nature" [Lan97]. To elaborate on this further, AL is the interdisciplinary study of living systems, taking the essential general properties found within them and adapting them for usage in man-made systems, as to synthesise AL-forms [Bed03].

There are many examples of AL being used within the entertainment industry. The early foundations of this being laid out by Reynolds in 1987, whose work was used as the basis for modelling avian creatures for many films [Rey87]. Reynolds work was further built upon in 1994 by Terzopoulos and his colleagues who modelled in great detail the behaviour of fish [TT94]. This included behaviours such as mating, feeding, learning, and predation. This work has went onto to be used in many short animated movies and video games.

Before Reynolds seminal work, there was a game called Little Computer People [Cra85], written by David Crane for the Commodore64 in 1985. It focused around a man and his dog, requiring a player to engage with him to satiate his needs or he would become unhappy or sick. The man would have a unique set of behavioural traits dictating his actions that would be generated from a serial number unique to each copy of the game. The game did not feature any learning but was diverse and novel enough at the time to garner enough attention to earn its place as a cult classic.

In 1993 we saw the release of Simlife [Max93], released by Maxis. This title was promoted as being inspired from AL research, featuring a simulated ecosystem, with variable terrain, climate, and a plethora of both plant and animal life. The game was packaged along with a notebook and it was suggested that the player take notes on their experience. The landscape featured a variety of geographic features and resources, which were distributed using cellular automata techniques. It featured intrinsic adaption, as it made no use of an explicit fitness function, creatures known as orgots would breed based on their own internal states, seeking a mate who is also ready to mate and within proximity.

### 2.3. The Creatures Series

One of the most notable and prominent early commercial successes being the Creatures series [Lab96], created by Steve Grand. The first Creatures game was released in 1996, with five more major releases from then leading up until 2001. The core gameplay relied on having players foster an emotional bond with creatures known as *norns*. They would raise and care for them as they gradually mature from child to adult, and eventually death. The artificial intelligence (AI) being somewhat flawed and slow to learn worked in favour of this series, giving norns a sense of believability and charm. Norns would explore their environment in a clumsy and goofy manner, eventually maturing into quite intelligent creatures capable of communicating with both each other and the player.

Norns made use of artificial neural networks, artificial biochemistry, and artificial evolution, allowing detailed simulations of both biological and neurological functions [CG99]. Norns could interact with their environment as well as each other with their primitive linguistic capabilities. The player is even able to talk to them through written word, allowing a user to teach their norns basic nouns and verbs through association. Their neural networks were capable of generalisation, taking information they learn from one scenario and applying it to a similar one [GC98].

The norns could sexually reproduce, using crossover and mutation to create offspring. It was possible for offspring to have more or less genes than their parents, meaning that the complexity of the norn's biochemistry and neural network could increase. Along with this, as norns could learn from one another this allowed for cultural transmission of knowledge. Meaning norns were capable of both biological and cultural evolution across multiple generations.

## 3. Methods

### 3.1. The Environment

For the sake of simplicity and performance the current implementation of the environment will be 2D and consist of a vector of tile objects (see Figure 1). These tile objects can contain game objects, such as spawners and food, as well as this they have an elevation variable which is used to generate the tile type. There are a variety of tiles to represent water (blue tiles), grassy areas (green tiles), mountains (grey and white tile), and so on. These different terrain types, as well as being aesthetically pleasing and easy for a user to visually interpret, also dictate where water sources are and where food spawners can be potentially placed, and thus how food is distributed across the map.

Procedurally generated environments made using octaves of simplex noise and redistribution methods were used as this allows a variety of landscapes to be made with ease. This also has the added benefit that the environment can be recreated using the seed value along with the other inputs, meaning it is not necessary to store each individual tile, only what game objects are contained on it. Food spawners will then be distributed across the map at their required elevation.

Food spawners have a set rate at which they will spawn their set food. There is a minimum and maximum radius that the food can be placed. The time is initialised to a random value on creation so they will not all spawn food at once. This will allow for agents to form a more natural distribution across the map and encourage migration once an area has been depleted of its currently available resources.

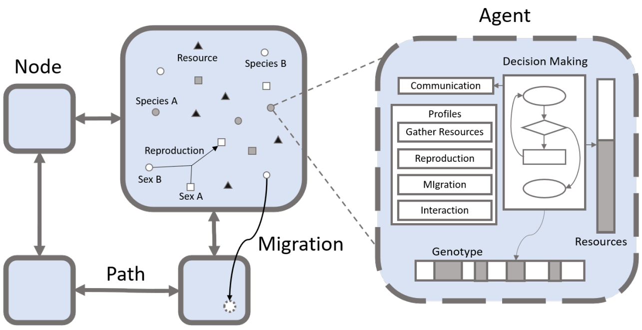


**Figure 2:** *Summary of CoEMAS to control non-player characters in a video game. With the map being split up into nodes (visually represented by tiles in Figure 1) an agent may choose to migrate to and from. As well as this, the internal processes of an agent is shown, from how decisions are made, and the linkage to its genome and current resources.*

### 3.2. The Agents

Each agent of the CoEMAS is composed of three main components (see Figure 2): physical laws, biological laws, and their genome. The physical laws are what restrictions are applied to an agents' movements and energy usage, as well as encompassing what interactions the agent can make with the environment and each other. The biological laws are what dictates how an agents' fitness is evaluated. Finally, within the genome is where all the information on the individual is stored, this has a direct effect on its behaviour, from how often it will eat, drink, sleep, and mate. As well as more complex behaviours such as if it exhibits separation and cohesion, the two behaviours found in flocking algorithms.

Each agent has a set of variables attached to them, these for the most part can be separated into state and will variables. State variables represent the current status of the agent, this includes its age, location. Will variables represent the agents current wants and needs, such as hunger, thirst, fatigue, and seeking a mate.

There are five distinct behaviour profiles: hungry, thirsty, sleep, mate, and migrate. These behaviours model the foundational physiological layer in Maslow's hierarchy of needs [McL07]. An agent's genome contains a threshold value for each of these behaviours, apart from migration which is defaulted to when no other profiles requirements have been met. To select a profile, a priority value for each profile is calculated using the relevant will variables and threshold genes. These values are placed into a vector in which the index of each priority coincides with the relevant profile, the index of the highest value can therefore be used to directly set the profile of the agent. This method reduces the number of necessary comparisons greatly and allows scalability as new profiles are added in future updates.

To quantise how these basic needs are met, a variable that represents an agents comfort is used. The comfort value will increase once the physiological needs have been met and will decrease when they are not. Once this meets a threshold value dictated within an agent's genome, they will then seek a mate. This value is a representation of an agent's well-being and mood. This also gives specimen that successfully manage their needs over a longer period an advantage over agents that are only sustainable within a short time frame. It is also important to apply a cost for creating offspring, as well as making the parents share a fraction of their resources with the offspring. This means that parents create offspring at a cost to themselves that may even incur death, thus creatures with high resource values will have an advantage in successfully breeding.

### 3.3. The Genome

As for an agent's genome, there are two main approaches to how these may be coded; high-level and low-level. High-level, wherein it is very clear what each genes purpose is and they may be represented as a variety of variable types. Whereas low-level represents the genome as a bit string, which is closer in line with how real world genetics work. The benefits of high-level are that it is easier to maintain and edit, each gene has a clear and distinct purpose and therefore can easily be edited and expanded upon.

Low-level, although more realistic, is not as flexible and easy to add onto as a high-level approach. As this system is intended to be used by an end user and edited to meet their needs, a high level approach should be used. A high-level approach would also in this case require less processing and memory requirements and as this is intended for implementation within a video game the ratio between efficiency and depth is important.

The genome contains a variety of variables, dictating when needs should be met, if they flock and ranges for flee and pursuing, their dietary requirements, as well as lifespan, and sight range. The more the genome can influence, the more complex the ecosystem can be. Each of these genes has a range that is allowed, along with the maximum amount a gene can change due to the creep operator.

### 3.4. Fitness Evaluation

The fitness in this case is how well an agent has adapted to live in its environment. If an agent survives and meets its base physiological needs it may be allowed to mate. When a creature is ready to mate though is will go through a process of evaluating potential mates in its vicinity. To allow for a greater degree of speciation a similarity bias has been used, agents will select other agents that are similar to it. A similarity exceeding 80% will incur a penalty of the excess squared, as to encourage variety in the offspring.

As well as similarity, the proximity of potential mates is also accounted for. The distance taking the available sight range into account is adjusted to a float value between zero and one, zero being a point at the maximum of their sight range and one being directly adjacent. This weighting of this half that of the evaluated similarity. This decision was made as agents would end up travelling large distances to find a mate that may only be a fraction of a percent better than one standing right next to it, which not only makes very little sense in the context of a natural system, but is also inefficient. As well as this, it allows separate flocks to remain separated from one another, while not ruling out breeding between them it will softly discourage it, this also encourages allopatric speciation to a degree.

## 4. Results

Table 1 shows the final ecosystem in a tabular format when the ecosystem's simulation starts with a population restricted to to apple eating species. All other genes of the agents were randomly generated. Food was made to be scarce in this example. The simulation ran for a long period of time. As a consequence breeding did not occur much and the population remained low. Unfit specimen were quickly weeded out.

Other behaviours were observed with different initial simulation parameters. For example, when food is abundant, we observed that the population grew much more and scavengers were predominant.

## 5. Conclusions and Future work

This early proof-of-concept has found very promising results already. With a very limited number of variables making up the genome of an agent there is already promising levels of speciation and interesting emergent behaviours being shown. The high-level approach to genes proved its merit, as the system went through stages of updates it was easy for new genes to be added on. By simplifying genetics down to only what is important it allowed for more focus and quicker development cycles than that of a low-level approach. This also means that in the hands of another developer, this tool would be much easier to understand and adjust to meet their needs.

**Table 1:** *Results of the ecosystem's simulation after being ran for a long period of time.*

| Lifespan | Hunger | Thirst | Fatigue | Mate | Comfort Inc | Comf Dec | Sight Range | Diet | If Flocks | Flee | Pursue |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 640065 | 3.65928 | 7.50893 | 3.11462 | 2.75093 | 0.00937189 | 0.00104682 | 106 | predator | 1 | 15 | 11 |
| 640065 | 3.65928 | 4.60109 | 0.186711 | 2.72696 | 0.00790425 | 0.00104682 | 89 | apple | 0 | 29 | 11 |
| 770798 | 3.63994 | 3.60134 | 5.82459 | 1.11729 | 0.00993204 | 0.00104682 | 192 | predator | 1 | 16 | 11 |
| 645823 | 3.65928 | 7.50893 | 4.67003 | 2.33567 | 0.0179379 | 0.00104682 | 89 | apple | 1 | 30 | 19 |
| 770798 | 3.65928 | 3.60134 | 5.83054 | 2.72696 | 0.00937189 | 0.00104682 | 89 | apple | 1 | 16 | 16 |
| 770027 | 3.65928 | 5.55611 | 5.83054 | 1.92212 | 0.00937189 | 0.00104682 | 99 | apple | 1 | 22 | 11 |
| 666096 | 3.65928 | 7.50893 | 5.83054 | 2.72696 | 0.0179379 | 0.00104682 | 89 | apple | 1 | 30 | 19 |
| 770798 | 3.65928 | 3.60134 | 5.83054 | 2.72696 | 0.00993204 | 0.00104682 | 106 | apple | 1 | 16 | 13 |
| 666096 | 3.65928 | 7.50893 | 5.83054 | 2.72696 | 0.0179379 | 0.00104682 | 89 | apple | 1 | 29 | 11 |
| 770798 | 3.65928 | 3.60134 | 5.83054 | 2.72696 | 0.00937189 | 0.00104682 | 89 | apple | 1 | 16 | 16 |
| 770798 | 3.63994 | 5.55611 | 5.83054 | 1.49821 | 0.00937189 | 0.00104682 | 99 | apple | 1 | 16 | 11 |
| 666096 | 3.61458 | 4.60109 | 0.186711 | 2.33567 | 0.00801543 | 0.00177799 | 89 | predator | 0 | 30 | 14 |
| 770798 | 3.65928 | 3.60134 | 5.83054 | 2.72696 | 0.00993204 | 0.00104682 | 106 | apple | 1 | 16 | 16 |
| 666096 | 3.65928 | 3.60134 | 5.83054 | 2.72696 | 0.0179379 | 0.00104682 | 92 | apple | 1 | 14 | 11 |
| 770798 | 3.65928 | 7.50893 | 0.186711 | 1.90082 | 0.0179379 | 0.00104682 | 137 | apple | 0 | 29 | 11 |
| 770798 | 3.65928 | 3.60134 | 5.83054 | 2.72696 | 0.00993204 | 0.00104682 | 106 | apple | 1 | 16 | 15 |
| 666514 | 3.66093 | 7.50893 | 5.83054 | 2.33567 | 0.0174479 | 0.00104682 | 134 | apple | 1 | 14 | 19 |
| 666096 | 3.63296 | 3.60134 | 5.83054 | 2.33567 | 0.0179379 | 0.00104682 | 89 | apple | 1 | 14 | 12 |
| 769814 | 3.65928 | 3.60134 | 5.83054 | 2.72696 | 0.00937189 | 0.00104682 | 106 | apple | 1 | 16 | 10 |
| 771678 | 3.65928 | 3.60134 | 5.83054 | 2.72696 | 0.00937189 | 0.00104682 | 89 | apple | 1 | 16 | 11 |
| 771678 | 3.65928 | 3.60134 | 5.83054 | 2.72696 | 0.0179379 | 0.00104682 | 92 | apple | 1 | 14 | 11 |
| 770798 | 3.65928 | 3.60134 | 5.83054 | 2.72696 | 0.00937189 | 0.00104682 | 106 | apple | 1 | 16 | 15 |
| 770798 | 3.65928 | 7.50893 | 5.83054 | 2.33567 | 0.00993204 | 0.00109254 | 89 | apple | 1 | 30 | 20 |
| 640065 | 3.63874 | 4.60109 | 0.186711 | 2.72696 | 0.0174479 | 0.00104682 | 115 | apple | 1 | 14 | 19 |
| 771678 | 3.65928 | 3.56596 | 5.83054 | 2.72696 | 0.0179379 | 0.00104682 | 92 | apple | 1 | 14 | 11 |
| 666096 | 3.65928 | 3.60134 | 5.83054 | 2.33567 | 0.00937189 | 0.00104682 | 89 | apple | 1 | 16 | 18 |

The agents were required to manage their needs, such as thirst and hunger, if managed correctly over an extended period their comfort value would increase. This value proved to be particularly useful, taking inspiration from the second and third layer of Maslow's hierarchy of needs, it attempts to quantify the general stability of a creatures living conditions as well as its mood. Once this value has met a threshold dictated within the genome, the agents would seek a mate. It was found that by considering both distance and genetic similarity in finding a mate, along with penalising those with too high of a similarity, allowed a diverse ecology to naturally emerge.

Along with this, enforcing a flat breeding cost along with the sharing of a quarter of the remaining resources from the parents to the offspring, necessitated the creatures to adapt their needs into a hierarchy to not only ensure they survive the breeding process, but their offspring also had a sufficient amount to begin life. The combination of these physical and biological laws allowed the agents to model an estimation of lifelike behaviour.

There are many areas within this project that could be expanded upon. At this point, layering more complex behaviours on top would not cost too much in terms of time, as the framework has already been built. A wider variety of plant food and the introduction of non-fixed food sources, such as mushrooms, which could spread across the landscape on the appropriate tile type. These more complex food types could even be given their own genetic structure, adding to the dynamic nature of the ecosystem.

Another area to work on in the future is the expansion of the senses, allowing for scent marking and more advanced speech functionality. A scent to one creature might signal a mate is nearby prompting it to follow the trail, but to another it may signal to avoid the area as it belongs to a predator. Species may also learn to mimic the scent of other creatures, this could lead to predators using scents to attract unwitting prey.

**Acknowledgements**

Some of this work is part of ***ANONYMOUS***'s final year dissertation submitted in partial satisfaction of the requirements for the Degree of Bachelor of Science in Computer Science.

**Acronyms**

**AI** artificial intelligence.
**AL** artificial life.
**CCEA** Cooperative Co-evolution Algorithm.
**CEA** co-evolutionary algorithms.
**CoEMAS** co-evolutionary multi-agent system.
**EA** evolutionary algorithm.
**EMAS** evolutionary multi-agent system.
**MAS** multi-agent system.
**NPC** non-player character.

**References**

[Bed03] BEDAU M. A.: Artificial life: organization, adaptation and complexity from the bottom up. *Trends in cognitive sciences 7*, 11 (2003), 505–512. 2

[CG99] CLIFF D., GRAND S.: The creatures global digital ecosystem. *Artificial Life 5*, 1 (1999), 77–93. 2

[Cra85] CRANE D.: Little computer people. [CD-ROM], 1985. 2

[DS08] DREŻEWSKI R., SIWIK L.: Agent-based co-operative co-evolutionary algorithm for multi-objective optimization. In *International Conference on Artificial Intelligence and Soft Computing* (2008), Springer, pp. 388–397. 2

[GC98] GRAND S., CLIFF D.: Creatures: Entertainment software agents with artificial life. *Autonomous Agents and Multi-Agent Systems 1*, 1 (1998), 39–57. 2

[JWSL13] JIAO L., WANG H., SHANG R., LIU F.: A co-evolutionary multi-objective optimization algorithm based on direction vectors. *Information Sciences 228* (2013), 90–112. 2

[Lab96] LAB C.: Creatures. [CD-ROM], 1996. 2

[Lan97] LANGTON C. G.: *Artificial life: An overview*. Mit Press, 1997. 2

[Max93] MAXIS: Simlife. [CD-ROM], 1993. 2

[McL07] MCLEOD S.: Maslow's hierarchy of needs. *Simply psychology 1* (2007), 1–8. 3

[Rey87] REYNOLDS C. W.: Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987), pp. 25–34. 2

[TT94] TU X., TERZOPOULOS D.: Artificial fishes: Physics, locomotion, perception, behavior. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (1994), pp. 43–50. 2