Agent and Evolutionary-based Modelling and Simulation of a Simplified Living System

Adrian Sośnicki
Velis Real Estate Tech
Cracow, Poland

Daniel Grzonka
Department
of Computer Science
Cracow University
of Technology
Cracow, Poland

Łukasz Gaża
Department
of Computer Science
Cracow University
of Technology
Cracow, Poland

KEYWORDS

Living Systems, Artificial Life, Agent Systems, Simulation, Modelling, Genetic Algorithms, Behavioural Systems.

ABSTRACT

Modelling artificial life has been an issue explored for several decades. However, science continues to surprise us with novel approaches to this problem. The aim of this paper is to innovative model a simplified living environment based on the agent paradigm and genetic algorithms. This paper also proposes a novel way of defining agent systems and artificial life embedded in a genetic approach. In the modelled and implemented environment there is one species of fauna and a simple species of flora that serves as food for the fauna. The fauna is implemented using agents inscribed in a genetic representation. The experimental part of the work includes calls to the simulator and the study of the dependencies resulting from the simulation mechanisms.

I. INTRODUCTION

Many technologies take their inspiration from the natural environment. This is certainly the right direction — nature has had billions of years to develop organisms adapted to the prevailing and often changing conditions. The method identified with adaptation is natural selection, which has helped form the genetic code of living beings for millennia. It now serves the broader scientific community. Virtual living beings, however, do not seek food or shelter, but function maxima or the shortest paths in a graph (i.e. the equivalents of optimal solutions).

The process of natural selection is somewhat like a random process, but the possible solution converges much more quickly towards the sub-optimal one, and adapts when the environment changes. A solution always exists — often very surprisingly — even for a complex and constantly changing environment.

This claim can be put to the test by constructing a simulator in which virtual entities try to survive. If the simulated biosphere is equipped with the right tools, natural mechanisms will allow it to find a way to develop and persist. Exploring such a virtual environment is an engaging activity whose solutions are often surprising. It shows how a genetic algorithm — and therefore low-level natural intelligence — creates high-level intelligence. And all this without prior knowledge of your own environment.

The rest of the paper is structured as follows. Section II discusses the related work. We briefly define the general agent paradigm in Section III. Section IV introduces genetic representation of simulated living system. Next section (V) presents formal model of our novel agent-based ecosystem. Evaluation of the proposed living system is presented in Section VI. Finally, Section VII concludes conducted experiments.

II. RELATED WORK

The topic of artificial life modelling is very widely covered in the literature. There are many publications devoted to the issues of agent modelling, as well as the construction of living systems environments. However, there is no universal standard, defining a general approach to solving the problem. A few selected works addressing these topics are presented below.

A fundamental example of an artificial life simulation is Conway's Game of Life, which was invented in 1970 by British mathematician John Conway and popularised by Martin Gardner [8]. It is one of the first and most famous examples of a cellular automaton. The game is played on an infinite plane divided into square cells. Each cell can be in one of two states: it can be either alive or dead. A dead cell with exactly three living neighbours becomes alive in the following unit of time. A living cell with 2 or 3 living neighbours remains alive; it dies with a different number of neighbours.

M. Ling in [14] describes the new Python library that allows for modelling of artificial life simulation and Digital Organism Simulation Environment (DOSE). The presented approach is based on GA and biological hierarchy. It starts from genetic sequence that create the whole population. Genetic code is based on the structure of 3-nucleotide codons in naturally occurring DNA, and is built from 3-character instruction set that accepts no operand In addition, the context of a 3D world consisting of ecological cells is introduced to simulate a physical ecosystem.

Another interesting solution is MUTANT: a multi-

Communications of the ECMS, Volume 36, Issue 1,
Proceedings, ©ECMS Ibrahim A. Hameed, Agus Hasan,
Saleh Abdel-Afou Alaliyat (Editors) 2022
ISBN: 978-3-937436-77-7/978-3-937436-76-0(CD) ISSN 2522-2414

agent toolkit for artificial life simulation by S. Calderoni and P. Marcenac [2]. It is generic platform that allows scientists in various fields of research to easily build simulation environment. The platform includes a model of self-adaptive agent with genetic evolving capabilities as learning mechanisms. It implements tools for agents design behaviour's programming environment's description and observation of running simulations. MUTANT is being developed in Java with the aim of being directly usable throughout the Internet.

J. Csonto et al. in [5] develop artificial life simulation based on multi-agent simulation system that could at least partially substitute the real experiments with real algae cells. Proposed simulator is based on real biological parameters of alga Chlorella kessleri, and use partial implementation of other mathematical models of algae population growth whereby it is possible to simulating the process of absorbing heavy metals from contaminated water. Model implementation is done in Swarm—multi agent object based simulation system and it's libraries.

There are also a number of review articles that attempt to provide a history and organise the nomenclature associated with Artificial-Life Ecosystems [7], characterise use cases for the idea of ALife (Artificial Life) [12], deals with the intersection of Artificial Intelligence (AI) and virtual worlds, focusing on AI agents and exploring the potential implications toward the human-level AI [17], representing a bottom-up approach to modelling complex life systems by using agents [15], or discuss major challenges to building live simulations covering various aspects [21].

III. THE AGENT PARADIGM

The idea of an agent paradigm (agent-oriented programming) dates back to 1990, with Yoav Shoham's publication of [20] dedicated to this issue. On the other hand, as early as in the 1970s, the roots of the software agent can be found in Carl Hewitt's actor model [1], [10]. Currently, the most popular definition of an agent is given by M. Wooldridge [24]:

An agent is a computer system that is embedded in a certain environment, and that is capable of autonomous actions in order to achieve ordered, specified goals.

Central to this definition is the feature of agent autonomy. It allows the agent to maintain full control over its internal state and the actions it takes [13]. Another feature is the perception mechanism, which allows the agent to observe the environment and the other agents in it. Based on the state of the environment and the state of other agents, it can make decisions [9].

An important element of an agent system is the environment. K. Cetnarowicz in [3] gives two types of components present in the environment — these are: (i) agents — a component of the environment, perceivable by the other actors in the environment; (ii) resources — components that do not have the ability to take initiatives, but can change according to their own established algorithm.

A formal notation of the abstract architecture of intelligent agents was proposed by M. Wooldridge [22], [24]. It defines an agent, an environment and functions representing the agent's various activities in the environment. According to the author, an intelligent agent is one that is capable of flexibly autonomous actions taken to achieve the goals [26]. By „flexible actions" he means three elements: (i) proactiveness – the ability to take initiative to achieve mandated goals; (ii) reactivity – the ability to perceive the environment and to react in good time to changes in it; (iii) social ability – the ability to interact with other agents (and even humans) to achieve mandated objectives.

As described earlier, agents are part of the environment in which they are situated. They can interact with it by performing certain actions. The environment reacts to these actions by changing its state. In[24] the following concepts and definitions concerning the environment and the agents that operate within it are presented.

Assume a finite set:

$$E = \{e, e', \dots\} \tag{1}$$

represents the set of states of the environment. A finite set of actions:

$$AC = \{\alpha, \alpha', \dots\} \tag{2}$$

represents the ability of agents to operate in the environment.

At each moment in time, the environment is in one of the states e, starting from an initial state e_0. Based on this state, the agent performs the appropriate action. As a result, the environment changes its current state to one of many theoretically possible states. This entire process can be referred to as the agent's movement (or history) and can be written as a sequence:

$$r : e_0 \xrightarrow{\alpha_0} e_1 \xrightarrow{\alpha_1} e_2 \xrightarrow{\alpha_2} e_3 \xrightarrow{\alpha_3} \dots \xrightarrow{\alpha_{u-1}} e_u. \tag{3}$$

The above discussion assumes that the environment is deterministic — so its state depends only on the history of actions taken by agents and changes to its own states. The environment may be non-deterministic, which means that the outcome of executing actions in certain states may be uncertain.

Formally, the environment can be defined by a triple:

$$Env = < E, e_0, \tau >, \tag{4}$$

where:
E — a finite set of environmental states,
e_0 — initial state of the environment,
τ — state transformation function.

If an agent does not refer to its experience when choosing an action, it is a purely reactive agent. It makes decisions based only on the current state of the environment. Such an agent can be defined by a function:

$$Ag : E \to AC. \tag{5}$$

In the context of this paper, agents should be considered in the context of reactive agents.

IV. GENETIC REPRESENTATION

A. Flora

A.1 Environmental map

Flora creates a map of the environment — a rectangular matrix that is also a map of the vegetation. It is navigated by agents. In the context of the taxonomy presented in this paper, the environment is accessible, deterministic, episodic, static and discrete.

A.2 Plant height

Each field on the map has a numerical value, ranging from zero to a specified maximum. A field with a value of zero is treated as barren land where nothing grows. A field with a value above zero is treated as having plants, which are food for the animals. In addition, if the plant is sufficiently mature, it will, in each cycle of the simulation, spread to one of up to 8 neighbouring fields with no plant.

The flora grows slowly at first. Its growth rate gradually increases and reaches a maximum when the plant is halfway to its maximum size. Then the rate of development drops until it reaches zero when the plant is fully grown. This is described by the function:

$$\Delta R = \frac{4\Delta R_{max} R(R_{max} - R)}{R_{max}^2}, \tag{6}$$

where:
ΔR — plant growth;
ΔR_{max} — maximum growth per cycle;
R_{max} — maximum plant size;
R — plant height.

The actual growth of the plant is in each cycle randomly between 50% and 100% of the value resulting from the function.

B. Fauna

The most important simulated living being is a representative of the fauna modelled as an agent based on a genetic algorithm, hereafter referred to in the paper as agent, individual or animal. It moves through the environment, feeds on the vegetation growing on it and reproduces. The simulated animals are hermaphrodites, so they need any partner to reproduce. This occurs instantly and results in one offspring.

B.1 Energy

The most important parameter of each agent is its energy — a characteristic based on the energy profile presented in [3]. This is a numerical value that regulates the animal's behaviour. If it is low, the agent will seek food which, when consumed, will be converted into energy. When it is high enough, the individual will be ready to reproduce. If it falls to zero the individual dies.

B.2 Chromosome

Each animal has a chromosome consisting of 40 bits. It defines six traits of an agent. The value of each trait is the decimal number decoded from the corresponding chromosome fragment plus one. Traits have a value of 1—16 or 1—256, depending on whether the gene is four-bit or eight-bit.

Eight-bit genes:

- Maximum Energy — The maximum level of energy an agent can have. The energy of an individual in the initial population or at birth is a fraction of the maximum energy;
- Life expectancy — The number of simulation cycles an animal will live before it dies of old age;
- Willingness to reproduce — An individual will only seek a reproductive partner if it achieves at least this much energy. This value is independent of the actual cost of reproduction;
- Wanted robustness of plant — An individual will only seek a reproductive partner if it achieves at least this much energy.

Four-bit genes:

- Visual range — Determines how many fields the animal can see around it, including diagonal fields. The agent can see the exact amount of food and features of other individuals within this range;
- Jaw size - Determines the maximum size of one portion of food.

In addition, each animal has a trait called fitness, with a value between 0 and 1. This value determines the closeness of the animal's genes to perfection. It is not a function of the evaluation of the individual, but only an intuitive measure of the quality of useful genes. It is calculated from the formula:

$$f = \frac{\sum_{i=1}^{n} \frac{G_i}{G_{imax}}}{n}, \tag{7}$$

where:
f — value of adaptation (fitness);
G_i — the value of i-this gene belonging to the adaptation genes;
G_{imax} — the maximum value of i–th gene belonging to the adaptation genes;
n — the number of genes belonging to the adaptation genes.

Selected genes make up adaptation. Individuals with a higher fitness value have an advantage over those with a lower one.

Fitting genes

Maximum energy – the higher it is, the more energy can be stored by an individual. It is also more likely that this value will be higher than the cost of reproduction and the desire to reproduce.

Life expectancy – long-lived animals exist longer in the environment, giving them more chances to seek food and reproduce.

Visual range – long range vision allows animals to move deliberately towards suitable food or partners.

Jaw size – this parameter determines the amount of food consumed during one cycle. A larger jaw size minimises the loss of valuable energy (feeding is an activity that costs energy) and saves time that the individual can spend searching for food or a partner.

Other genes

Willingness to reproduce – this gene defines a threshold value of energy for an individual to decide to reproduce. A low value will therefore encourage the individual to reproduce at a low energy level. This will accelerate population growth and gene transfer. But too low will expose the individual to life-threatening low energy levels shortly after reproduction. A high value will allow individuals to store more energy, giving them a safe energy reserve just after reproduction.

Wanted robustness of plant – if an animal seeks out a plant of low maturity, it will widen the range of food available to it, giving it an advantage in the population. This naturally increases the animal's energy, allowing for rapid reproduction and therefore a rapid increase in population size. On the other hand, a high value will allow vegetation to grow faster on the map and increase the chance of spreading in barren fields.

B.3 Algorithm of agent behaviour

Individuals act towards a simple, predetermined algorithm. An agent's behaviour is governed by its current energy value, its desire to reproduce, and the food it seeks. Depending on these values, the individual performs a different action.

Moving – the animals move around the map in eight directions. The movement takes each of them a whole cycle and they then cover a distance of exactly one field. Moving costs energy.

An agent moves in a particular direction if there is an object it is aiming at in its line of sight. This object is food, when the individual wants to eat, or a potential partner, when the animal wants to reproduce. The agent always moves towards the nearest object that meets its requirements. If there are more such objects, it will move towards a randomly selected one. If there is no object in sight, the agent will move randomly.

Reproduction – when an animal wishes to produce offspring, i.e., enters a state of reproduction, it checks the adaptation value of other individuals in sight. It sees only those agents that are also in the state of reproduction and whose adaptation value is sufficiently high. It must be at least as high as the agent's own adaptation minus some tolerance. In this way individuals will not interbreed with others who have received an unfavourable mutation.

The animal will move towards the nearest suitable partner to be in the same field as it. In case there are many partners available in the field, one will be chosen randomly. In the final step, the animal must be accepted — it must have a high enough adaptation value from the partner's point of view. Otherwise, the suitor will be ignored and will have to wait out the simulation cycle without doing anything. If the courtship is successful, the parents pay the cost of reproduction in energy. They also both go into an idle state to prevent them from reproducing again in the same simulation cycle. Exactly one offspring is born, in the same field as the parents. It is fully independent from the moment of birth.

Feeding – if the individual does not have enough energy to reproduce, it will search for food. It heads towards a field where there is sufficiently tall vegetation. The agent pays the cost of feeding itself. An agent during one cycle can eat as much as the size of its jaw. An agent eats less food if its energy requirement is lower, or if the plant is too low. The value of the map field — and thus the plant growing on it — will be reduced by the value eaten by the animal. If it is reduced to zero, the field becomes barren.

V. AGENT MODEL

As stated by D. Grzonka in [9], the constituent elements of agent systems can be: types of agents, locations of agents, strategies implementing coherent goals, executable actions, states of the environment and operators allowing to perceive and interact with the environment. Based on the definition proposed by the author, a definition of a multi-agent system inscribed in the problem under consideration can be presented:

$$MAS = \{AG, ID, TP, LOC, K, ES, ACT, ST, GL, GEN\}, \tag{8}$$

where:
AG — a set of agents belonging to a multi-agent system;
ID — a set of unique identifiers for agents;
TP — a collection of all agent types;
LOC — a set of locations where agents may be present;
K — a set of all possible states of knowledge of the agent;
ES — a set of all possible states of the environment;
ACT — a set of actions that can be performed by agents within the environment;
ST — a set of strategies implemented by agents;
GL — a set of agent objectives;
GEN — a set of all possible combinations of features recorded in chromosome form.

An agent (ag) is defined as follows:

$$AG \ni ag = \{id, tp, st, k, l, gn, gl, en, ae, \gamma, \beta, \delta, \alpha, \epsilon\}, \tag{9}$$

where:
$id \in ID$ — a unique system-wide identifier for the agent;
$tp \in TP$ — agent type;
$st \in ST$ — agent's strategy;
$k \in K$ — current (temporary) knowledge of the agent;
$l \in LOC$ — current location of the agent (position on the map);
$gn \in GEN$ — a set of agent characteristics recorded in chromosome form;
$gl \in GL$ — agent's current objective;
$en \in \mathbb{Z}$ — agent's current vital energy;
$ae \in \mathbb{N}$ — current age of the agent;

γ — observation (perception) function to monitor the state of the environment;
β — strategy selection function;
δ — a decision-making function which, on the basis of the strategy pursued, selects actions;
α — an action function that, based on a selected action, executes it on the environment changing its state;
ϵ — agent adaptation function.

A detailed definition of the model components can be found in [9].

The paper proposes one type of agents: $TP = \{Ag_{fauna}\}$, and 2 strategies. Strategies define how the fauna–agent functions in the environment and how the goals are achieved. Strategies can be written using a set:

$$ST = \{st_1, st_2\}, \tag{10}$$

where:
st_1 — prospecting for food;
st_2 — seeking a reproductive partner.

Strategies are selected depending on the agent's objectives:

$$GL = \{gl_1, gl_2\}, \tag{11}$$

where:
gl_1 — increase energy levels;
gl_2 — reproduction.

The most essential element of agents, which defines their causal capabilities, are actions. In the model under consideration this will be a set consisting of 8 actions:

$$\begin{aligned} ACT = \{&choose_{fauna}, choose_{flora}, move_{towards}, \\ &move_{random}, eat, reproduce, idle, die\}, \end{aligned} \tag{12}$$

where:
$choose_{fauna}$ — selecting a location within the agent's line of sight where there is a suitable reproducing partner;
$choose_{flora}$ — selecting a location within the agent's perception range where the flora corresponding to the agent is present;
$move_{towards}$ — movement of an agent towards a location selected by one of the actions *choose*;
$move_{random}$ — movement of the agent in a random direction in case the action *choose* does not find a suitable location;
eat — eating flora to recover energy;
reproduce — creation of a descendant;
idle — idle action;
die — killing the agent.

Within a single simulation cycle, within a selected strategy, an agent may perform one or more of the actions listed above.

VI. EXPERIMENTS

This experiment focuses on the development of the agents' features during the course of the simulation.

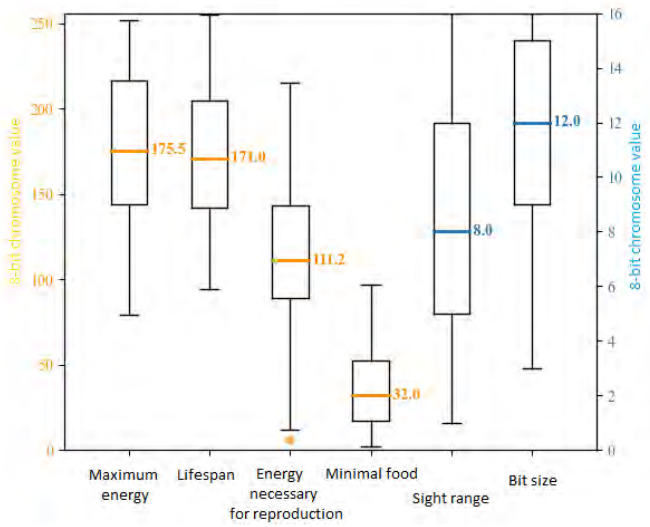


Fig. 1: Comparison of gene values at local population maxima of multiple simulation calls

Parameter	Value
Map dimensions	10x10
Number of agents in initial population	30
Maximum number of interactions	10000
Reproduction cost	50
Cost of inactivity	1
Cost of movement	3
Cost of feeding	1
Adaptation tolerance rate of partner	0.15
Agent's initial energy at birth	30% of max
Number of chromosome crossover points	1
Mutation chance	0,1%
Maximum flora height	100
Maximum flora growth per cycle	5
Height of florets allowed to reproduce	70% of max
Size of new rhizome after flora reproduction	1
Initial map filling with flora	50%

TABLE 1: Simulation parameters

Every fixed number of cycles, numerical values of faunal traits are recorded, from which box plots will be created. All simulations were repeated 100 times. At the end of each simulation the number of cycles it took is read out. After each simulation has been called enough times, a box plot is produced indicating how long the population persists on a given set of parameters.

For each simulation called in the comparison study, the median value of each gene from the entire simulation run is recorded. After multiple simulations, a box plot of typical gene values for a given parameter set is produced.

Figure 1 presents the distribution of typical gene values over multiple simulations on the example parameter set shown below (see: tab. 1).

At the initial stage, the animals have a wide variety of gene values, as can be inferred from the zero-cycle

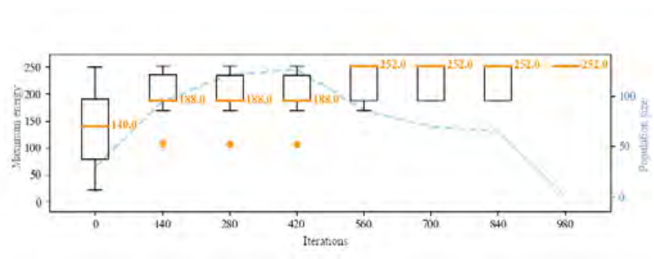


Fig. 2: Graph of faunal maximum energy gene values

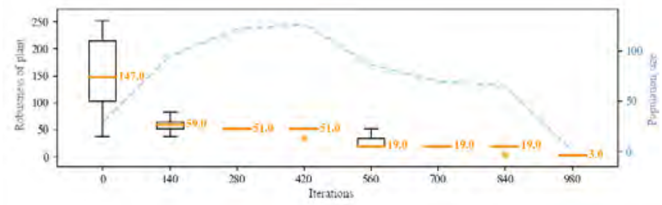


Fig. 7: Graph of the gene value for the fauna's wanted robustness of plant

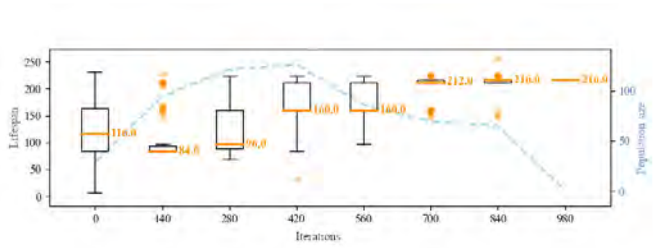


Fig. 3: Graph of fauna life expectancy gene values

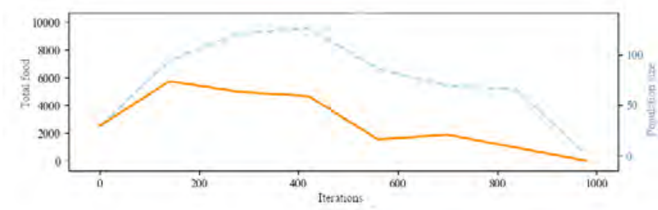


Fig. 8: Graph of total food intake

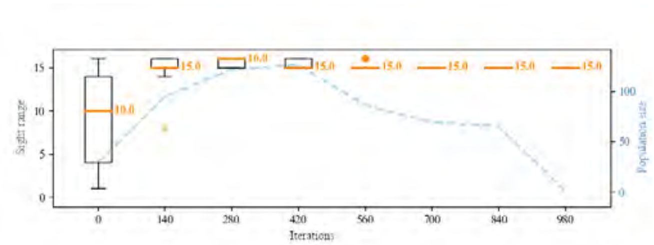


Fig. 4: Graph of fauna sight gene values

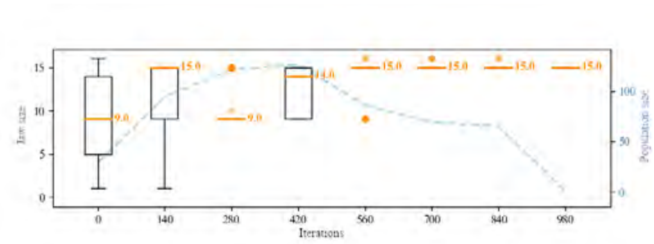


Fig. 5: Graph of fauna jaw size gene values

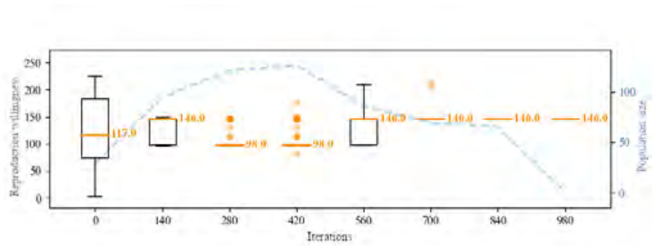


Fig. 6: Graph of faunal reproduction willingness gene values

graphs in Figures 2 – 7. The ranges of the graphs tend to shorten as the faunal population increases. Animal fitness, which can be observed in Figure 9, increases with successive cycles. Agents become more and more perfect, fitness gene values increase until the population dies out.

Due to the large spread of traits, the initial population consists of individuals able to thrive in the environment and those whose gene set will prevent them from surviving. The latter group dies out in the first few dozen cycles of the simulation. The former feeds and begins to reproduce. After some time, the population begins to consist only of individuals that are able to live and reproduce. There is a significant increase in the population. Soon, however, there begins to be a shortage of food. As can be seen in Figure 8, population size is strongly correlated with available food.

Genes that are scored as significant for fitness tend to rapidly optimise upwards. An exception is the eye range gene, whose graph is shown in Figure 4. For the 10 by 10 field map, a visual range of more than five (which allows an individual to observe the entire map from the centre) was not significant for survival.

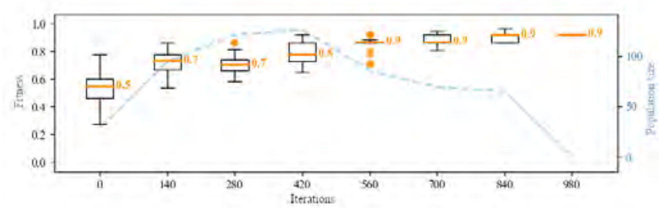


Fig. 9: Graph of faunal fitness values

VII. SUMMARY AND FUTURE WORK

In this research, a simulator of a simple natural environment inscribed in the agent paradigm has been implemented. The operation of the simulated environment was described and a formal multi-agent model of the problem was proposed. A number of experiments have been conducted, which have allowed several hypotheses to be made. The experiments made it possible to find the reason for the extinction of the simulated fauna. It was, in a way, a side effect of animal adaptation. Individuals eating lower and lower plants and enlarging their jaw gained an advantage in the environment. This combination ultimately depleted the environment of resources every time.

The simulations carried out have shown that excessive model parameterisation can adversely affect the results obtained. The developed simulator does not exhaust the whole issue of modelling natural environments. As part of future in-depth research, it would be worth extending the model to include other classical factors presented in literature and widely consolidated, such as the sex of individuals, the existence of predators, a more complex plant model (e.g. edible fruit), extended needs of individuals (e.g. sleep) or diverse terrain topography with different movement costs. It would also be useful to describe the simulator itself in more detail in future work.

References

[1] R. Atkinson and C. Hewitt. Synchronization in actor systems. In *Proceedings of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, POPL '77, page 267–280, New York, NY, USA, 1977. Association for Computing Machinery.

[2] S. Calderoni and P. Marcenac. Mutant: a multiagent toolkit for artificial life simulation. In *Proceedings. Technology of Object-Oriented Languages. TOOLS 26 (Cat. No.98EX176)*, pages 218–229, 1998.

[3] K. A. Cetnarowicz. *Problemy projektowania i realizacji systemów wieloagentowych.* AGH Uczelniane Wydawnictwa Naukowo-Dydaktyczne, 1999.

[4] K. A. Cetnarowicz. *Paradygmat agentowy w Informatyce: Koncepcje, podstawy i zastosowania.* Akademicka Oficyna Wydawnicza EXIT, 2012.

[5] J. Csonto, J. Kadukova, and M. Polak. Artificial life simulation of living alga cells and its sorption mechanisms. *J. Med. Syst.*, 25(3):221–231, jun 2001.

[6] Y. Demazeau and J.-P. Müller. Decentralized artificial intelligence. In Y. Demazeau and J.-P. Müller, editors, *Decentralized A.I. : Proc. of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Cambridge, England*, pages 3–13. North-Holland, 1990.

[7] A. Dorin, K. B. Korb, and V. Grimm. Artificial-life ecosystems - what are they and what could they become? In *ALIFE*, 2008.

[8] M. Gardner. Mathematical games. *Scientific American*, 223(4):120–123, 1970.

[9] D. Grzonka. Inteligentne systemy monitoringu procesów harmonogramowania w rozproszonych środowiskach dużej skali w ujęciu wieloagentowym. *Praca doktorska*, 2018.

[10] C. Hewitt, P. Bishop, and R. Steiger. A universal modular actor formalism for artificial intelligence. IJCAI3. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, pages 235–245, 1973.

[11] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, Jan. 1998.

[12] K.-J. Kim and S.-B. Cho. A comprehensive overview of the applications of artificial life. *Artificial Life*, 12(1):153–182, 2006.

[13] M. Kisiel-Dorohinicki. *Agentowe architektury populacyjnych systemów inteligencji obliczeniowej*, volume 269 of *Rozprawy. Monografie.* AGH Uczelniane Wydawnictwa Naukowo-Dydaktyczne, 2013.

[14] M. H. T. Ling. An artificial life simulation library based on genetic algorithm, 3-character genetic code and biological hierarchy. 2012.

[15] C. M. Macal. *Agent-Based Modeling and Artificial Life*, pages 1–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[16] Z. Michalewicz. *Algorytmy genetyczne + struktury danych = programy ewolucyjne.* Wydawnictwa Naukowo Techniczne, 1999.

[17] V. M. Petrović. Artificial intelligence and virtual worlds – toward human-level ai agents. *IEEE Access*, 6:39976–39988, 2018.

[18] J. Rocha, I. Boavida-Portugal, and E. Gomes. Introductory chapter: Multi-agent systems. In *Multi-Agent Systems.* IntechOpen, 2017.

[19] S. J. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 1994.

[20] Y. Shoham. Agent oriented programming. technical report stan-cs-90-1335. *Computer Science Department, Stanford University*, 1990.

[21] S. Swarup and H. S. Mortveit. Live simulations. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, page 1721–1725, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems.

[22] G. Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* The MIT Press, 1999.

[23] G. Weiss. *Multiagent Systems.* The MIT Press, 2013.

[24] M. Wooldridge. *An introduction to multiagent systems.* John Wiley & Sons, 2009.

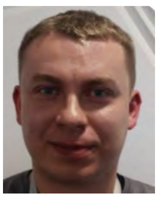
[25] M. Wooldridge and P. E. Dunne. *The Computational Complexity of Agent Verification*, pages 115–127. Springer Berlin Heidelberg, 2002.

[26] M. J. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(2):115–152, 1995.

AUTHOR BIOGRAPHIES

ADRIAN SOŚNICKI received his B.Eng. and M.Sc. in the field of Computer Science at Cracow University of Technology, Poland, in 2016 and 2020, respectively. Currently, he is Senior IT Business Analyst / Quality Assurance in Velis Real Estate Tech, Cracow. His main areas of interest include software engineering (programming, databases), project management and business analytics. If you want to contact him, please send an e-mail to adrian.sosnicki.skrzynka@gmail.com

DANIEL GRZONKA received his B.Eng. and M.Sc. degrees from Cracow University of Technology, Poland, in 2012 and 2013, respectively. In 2019 he received his Ph.D. degree from the Polish Academy of Sciences (in cooperation with Jagiellonian University). All degrees (with distinctions) are in Computer Science. Currently, he is an Assistant Professor at the Cracow University of Technology and Vice-Dean for Education at the Faculty of Computer Science and Telecommunications. He is also the laureate of the prestigious START 2019 competition of the Foundation for Polish Science for the most outstanding young scientists. The main topics of his research are monitoring systems, grid and cloud computing, multi-agent systems, task scheduling problems, data mining and high-performance computing. For more information, please visit: www.grzonka.eu



ŁUKASZ GAŻA received his M.Sc. in the field of Applied Physics with Computer Modelling at the Tadeusz Kosciuszko Cracow University of Technology. Since 2019 he is a Research and Teaching Assistant at the Tadeusz Kosciuszko Cracow University of Technology. His e-mail address is lukasz.gaza@pk.edu.pl