Preprint of: Balabanov, K., Cejrowski, T., Logofătu, D. et al. Evolving Systems (2019). https://doi.org/10.1007/s12530-019-09298-1

Noname manuscript No.
(will be inserted by the editor)

Study on Population Dynamics for Triple-Linked Food Chain Using a Simulation-Based Approach

Kristiyan Balabanov · Tymoteusz Cejrowski · Doina Logofătu · Costin Bădică

Received: date / Accepted: date

Abstract The procedures based on simulation have become a feasible testing method that does not require investing valuable resources to create a concrete prototype, especially with the increasing computational power of computers. Thus, design changes can be adopted and design errors can be fixed before it is too late. Simulation turns to be a cheap, safe and often more acceptable from an ethical perspective. In our work we summarize the results from the analysis with the help of a computational simulation of an elementary, yet analytically intractable problem scenario from the field of ecology. Our main goal is to confirm that even with a seemingly simple agent-based model and simulation, one could obtain plausible results regarding a system's real life behavior. As a last point, we propose an efficient alternative for analysis, rather than the expensive simulation process.

Keywords Population dynamics · Ecological simulation · Agent-based modeling · Predator-prey relation · Triple-linked food chain · Evolution-inspired optimization.

---

1 Introduction

Simulation is a a well-known and wide-spread method of system analysis and fault diagnostics. A lot of different fields of science make use of it due to the various advantages that it could offer. As an example, it is more economical to test the aerodynamics of a car without the need of an actual prototype [28]. On the other hand, in some projects it is simply the most suitable approach with respect to effort-cost ratio, e.g. to simulate a space rover's mission to another planet [36]. Simulation can be also used as a prediction tool for an assessment of the effects within possible climate catastrophes [25]. Furthermore, a well-designed simulation can not only save time and money, but also give a researchers chance to explore scenarios which are to hazardous to be enforced in natural environment. These kinds of ecological simulation has become quite popular recently due to the numerous environmental issues. For example, according to [34] one-third of the plant and animal species in the United States are at risk of extinction. Species are becoming endangered because of reckless management of the Earth's resources, indirectly caused by globalization and industrialization. Having such situation bounds to affect predator/prey chains in the respective ecosystems. With proper simulation methods one could analyze the consequences of the resulting imbalance without involving actual animals.

In this paper we present the results we gathered when simulating a simple marine ecosystem on a micro level. We based our model on the one described in [5], but extended and formalized it according to the well known design of cellular systems and more precisely a *cellular automaton.* The main goals that we set for this project were to further confirm the statement introduced in [5] that even somewhat complex real world phenomena/events/scenarios can be simulated with a relatively simple abstract model, that, nevertheless, exhibits authentic behavior. Our emphasis, however, was on proving the strengths of computational simulation when considering analytically intractable problems. The tests that we present served not only the purpose to verify the accuracy of the extended model, but also to deliver useful insights regarding the problem under consideration, such as input/output relation, without the need of a complex formal description. Furthermore, we acknowledge the pitfalls of simulation-based analysis in terms of computational effort needed and shortly discuss a more efficient approach to optimize the given problem scenario with the use of an *evolutionary programming* technique.

The design of the application follows an established model of a simulation as described by [13]. Essentially it involves a system's model, some inputs used in the model and the respective outputs produced by the model (refer to Fig. 1). In this matter of thought a model is simply an abstraction of a real world phenomenon or environment, which is too complex to theoretically describe in detail. Thus, the model is by no means perfect or accurate and its level of abstraction determines the degree to which the simulation's results would resemble the outcome of the experiment should it be conducted in the real world. Naturally, the higher accuracy of the model comes at a cost, but experience in the field has confirmed that often as not even low level abstractions can yield plausible results. The input is usually some initial state of the system and its components. The unknown variable in a simulation, i.e. the subject of interest, is the output produced by inserting the input into the given model. It is rather a relatively simple scenario in which the model description is known. Although, this is not feasible in many cases. Currently, researchers focus on developing complex models for dynamic and non-stationary systems. In such systems, the model is unknown and being estimated by use of universal functions (artificial neural networks, neurofuzzy networks) whose parameters can change over time. The area of algorithms and solutions addressing that problem has been named as evolving intelligent systems (eIS) [2]. In our work we provide fixed, ready to use description of the model. We focus on analyzing simulation output rather than searching universal approximations.

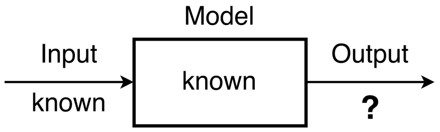

Fig. 1 The structure of a typical simulation problem as given by [13], page 9.

A variety of problems can be successfully addressed with simulation means, which are typically more economical (e.g. when testing the aerodynamics of cars [28]), more feasible (e.g. simulating a space rover's mission to another planet as described by [36] or when the rate of the observed phenomenon is too fast/slow in real-time as is the case with plant growth or explosions) or ethically more acceptable (e.g. performing air-bag tests or most experiments involving animals). Dewdney's proposal is a typical example of all three and more precisely the study of an ecosystem. This topic has gained popularity in the recent years as the globalization and industrialization have taken their tolls on the environment and the consequences have become evident. Species are becoming endangered or extinct and while environmentalists are coping to mitigate the damage, scientists are more concerned with determining the potential effects of the caused imbalance or lack of a species on the remaining ones. In other words, how would the predator/prey chains in an ecosystem change [7].

2 Problem Description

In its core the problem scenario used for the experiments is a relatively simple one: a finite space populated with objects of certain type, that interact with one another according to predefined rules. A less abstract extension of this scenario would be any ecosystem and the species populating it. Following research done in [41] biodiversity can enhance ecosystem services and temporal stability. On the other hand according to [12] equilibrium feasibility vanishes in species rich systems. In our experiments we modeled a sea ecosystem and three marine species forming a food chain. For the concrete design we used the well-known Wa-Tor world described in A. K. Dewdney's work [10]. It is a torus-shaped world with no landmass, but only one great ocean inhabited by marine species. To simulate an actual ecosystem, each species has a specific behavior according to which it interacts with both the environment and members of other species'. The behavior consists of a set of predefined rules, which reflect the real world equivalents of the respective species' to a certain degree. All rules are defined on a micro level, i. e. only an individual's current state and its direct neighbors are considered [5,29,30]. Each rule can be easily modified to reflect any introduced change of both internal (evolution) or external (cataclysmic event) nature, as it happens in the real world, and then the potential impact on the environment can be observed. Originally Dewdney described a world inhabited by only two species forming a linear predator-prey relationship — sharks (predator) and fish (prey). Their interaction was pivotal for the environment's fate: sharks can survive only if there are enough fish to hunt, and fish can survive only if there are not too many sharks to hunt them to extinction. Fish feed on infinite plankton, which is not modeled for the sake of simplicity. Following the example of [5], we extended Dewdney's idea with an additional third species to create a slightly more complex food chain — whales, which hunt both sharks and fish, but have no enemy. This new species is partially theoretical as it does not depict the actual diet of real world whales.

Such a scenario related to ecology is a perfect example of a problem domain, that is well-nigh impossible to solve using real life resources for two major reasons: a) constructing and populating or isolating an actual ecosystem would be insanely expensive, unless in a very small scale, which might not yield the desired authenticity and/or results; b) experiments with animals in general raise a lot of questions regarding moral principles and are often reproached by modern society. On the other hand, formally specifying a mathematical model of the problem would prove to be as equally difficult simply due to its complexity — thousands of agents, each with its own characteristics, dispersed within the environment, randomly roaming and interacting. In other words, the given problem, although computationally solvable with finite resources in theory, can be seen as analytically intractable in practice. Therefore, developing a computer simulation using a moderately complex model is a viable approach to obtain valuable insights regarding the problem scenario, albeit not necessarily the optimal solution [43,39,15]. In our work we strove to confirm the accuracy of the model presented by [5] and its ability to self-sustain, but also tried to determine whether a relation between the input parameters and the ecosystem's ability to self-sustain exists. Moreover, we searched for an approach to efficiently find optimal input parameter combinations, i. e. such that do not cause an ecosystem collapse (species becoming extinct).

3 Related Work

In last years there was a couple of projects which used a Wa-tor concepts as the basis of scientific projects. Studies presented in [26] compare two models against real-life example for hare and lynx population dynamics. First model was based on differential equations from widely used Lotka-Volterra model [37] and the second was the Wa-Tor model. Both describe the predator and pray relationship: hares with lynxes and fishes with sharks. It was shown that the Wa-Tor results resemble the differential equation model but also go further through spatial distribution. According to [40] changes in the dispersion pattern should be considered alongside changes in size when interpreting population dynamics. The overall population fluctuations of Wa-Tor model suits better to real-life example charts than the differential equations do. One of the conclusions was that modeling does not necessarily require a top-down differential equation model to provide realistic results.

Authors in [4] present general framework for modelling and simulation continuous dynamic systems with Belief-Desire-Intention paradigm applied and use of Jason AOP language. Each agent was described as a software module that provides an interface with the external world and consists of three components: belief base, plan library and reasoning engine. For continuous systems models are defined with help of differential equations and such approach implies use of numerical integration methods which are described in the paper. Predator-pray dynamic system experiment was conduced based on triple node graph-based model. Population dynamic was recorded and conclusion about robustness and efficiency of various numerical integration methods were drawn.

Discrete Wa-Tor model can be used as a starting point for more complex ecosystems. For example, the field of game industry is one where advanced simulations broadly occurs. In order to make a game-play realistic, digital characters should provide more sophisticated behavior than a basic if-else interaction. For game designers and programmers it is crucial to model artificial behaviour in a way where a character state is somehow reflected by the closest neighbourhood. For these reasons, researchers try to develop new approaches based on neural networks. The test driven development imply ideas where the need of an efficient testing framework broadly occurs. Wa-Tor was chosen as the one in [32] where authors build a simple percepton model in order to describe the internal emotion of a character. With the help of such a predefined model, they tested it on predator-prey architectures like Wa-Tor and similar ones.

Another promising approach for ecosystem modeling will be use of flocking behaviour [38]. This term was introduced by Craig Reynolds and describes the set of rules how the individuals would move within a space. In such scenario, modeling do not center upon the predator-prey relationship but more about movement of individuals called boids. According to Reynolds and his explanation there are three main rules describing individuals movement: separation (steer to avoid crowding local flockmates), alignment (steer towards the average heading of local flockmates), cohesion (steer to move toward the average position of local flockmates). With this kind of defined rules, one could possibly model the fish behaviour and adapt such sophisticated movement rules to the Wa-Tor ecosystem.

4 Implementation Details

As more than 30 years have passed since Dewdney's initial publication, one can easily find various implementations of the Wa-Tor scenario. Wa-Tor fundamental concepts focuses on an environment with only two species and give a researcher a general description of the interaction rules between them. To test and verify our ideas one has to find a tools which allow to redesign and interact with researcher's proposed model. Such software should be easy to use and be described as an open source project. Currently there is a only a few software implementations of Wa-Tor simulation which meets the requirements.

Cute simulations and a sub-project called *Wa-Tor simulation* [33] is one of the most recent projects dealing with Wa-Tor. This is a simple and portable Java application implementing two species: sharks and fishes. One can easily define input parameters such as the environment size, percentage space-fill of specific species, simulation length and time for reborn. Such a simply model could be a good starting point for further research, but currently the project has no support for more complex models and a scientific analysis. A similar one, the *Wa-Tor* android app [14], was developed for simulation analysis only in scope of testing the basic concepts. Also it is not suitable to verify presented ideas. Several available projects can be found - they are commercial, non-scientific or outdated due to the framework support.

To test our ideas we had to developed a simple and robust software application. Proprietary software was excluded from consideration. After a brief evaluation of the *open-source* solutions according to the criteria *maturity* (how well developed is the product so far), *longevity* (what are the prospects of the software being further supported and developed) and *flexibility* (effort needed to integrate/modify the product) as suggested by [36] showed that neither was adequate enough, hence our initiative to design and implement a new set of tools using Java. Lack of tools to simulate three or more species was also a major point for proving the need to design one's own solution. Developed application for *Wa-Tor simulation* was designed with focus on proposed *cellular architecture*, more detailed in next section. The main component of the program is a cellular board of a defined size where individual cells are considered to be one of the three species from *Wa-Tor simulation*. Cellular architecture is reflected in classes corresponding to a given species. For example, the objects of *Fisch.java* class will be a cell for fish species with defined interaction rules. The two genre version of the program was tested for correctness of the simulation against solutions that were similar in the rules of interaction.

4.1 Application Architecture and Representation

For the design of our application we chose a classic bio-inspired approach: an abstract cellular system. The idea is derived from biological tissues, where the fundamental unit is the cell. The cell is, indeed, a quite complex structure by itself [22], but when multiple cells cooperate even on a very basic local level, the outcome is a multicellular organism with unmatched capability compared to that of the individual cells building it. The human body is just one of countless examples in nature. This approach has already been widely used in design leading to the generalization of the resulting system type into the so called cellular systems. One among many precisely described in [31] is a cellular system describing the interactions between a growing tumor next to a nutrient source and host's immune system. Essentially, a cellular system is a finite collection of basic units building a space. The units can be called cells and the space an organism accordingly. Each cell is identified by specific information about itself (e. g. an n-tuple of numerical values) and this information at a given time is called a *state*. The dynamics of such a system are expressed with the change in the cells' state depending on various factors. This *state transition* can depend on the current state of the cell, its past state and/or on the states of the surrounding cells. The collection of cells that can directly influence each other's state is called a *neighborhood*. Once the system is running, the cells update their states over time according to a predefined set of state transition rules and always reside in one from a finite set of possible states. A more detailed description of cellular systems is offered by [16, chap. 2].

The most prominent advantage of cellular systems is that they offer a simple modeling approach on a micro level, which however, can still yield plausible insights regarding the global behavior of the model. In the case of our problem, it is much easier to model the behavior of separate individuals based on their direct neighbors, rather than model the behavior of the entire population as one complex object.

From the various kinds known today we adopted the relatively simple, but popular *cellular automaton* (CA), more precisely the 2-dimensional *game of life* CA, as the base architecture of our application. A vivid example is John Conway's Life Game [19,20]. Using Floreano et al.'s [16] decomposition of a cellular automaton into its components and their elaborate description, we modeled the Wa-Tor ecosystem scenario as a cellular automaton as follows:

*Base unit.*The base unit is the cell represented as a square grid tile of certain color in the cellular space.

*Cellular space.*The space is a 2-dimensional lattice of cells forming a rectangular grid with size 500 × 300 cells.

*Time variable.*The state transitions in the system unfold along a discrete time axis with *cycle* as its base unit. One state transition happens per cell per cycle.

*State and state set.*The state of each cell consists of the animal type that the cell represents together with its respective attributes tuple. Therefore, each cell can reside in one of four major states

$$\{Fish,\ Shark,\ Whale,\ Emptycell\}$$

and a multiple of sub-states defined by the numerical values of the respective species' attribute tuple.

*State transition function.*The transition from one state to another is represented by the actions that an individual can perform, such as moving, feeding, reproducing and dying. Both the major state of the neighbor cells and the instance of the attribute tuple of the considered cell influence the transition. It is not deterministic (e. g. a fish surrounded by empty cells could move to any one of it), and where more choices exist, their probability distribution is uniform.

*Neighborhood.*The *von Neumann* neighborhood is implemented, i. e. a cell's state is directly influenced only by neighbors at a *Manhattan distance* of 1 from it (its direct upward, downward, leftward and rightward neighbors).

*Boundary conditions.*For practical reasons the cellular space cannot be infinitely large, hence the need of appropriate boundary conditions to ensure that the system has a homogeneous neighborhood (every cell has the same type of neighborhood). We chose *periodic* boundary conditions by connecting opposite ends of the grid and essentially eliminating the boundaries, i. e. transforming the 2-dimensional grid into a 2-dimensional toroid (torus).

*Initial conditions.*The size of each species' population as well as the initial numerical values for each individual's attribute tuple (equal among all individuals of the same species). Both can be selected by the user before the start of the simulation. The distribution of the individuals over the cellular space is uniform.

*Stopping conditions.*The simulation stops after 3000 cycles (user adjustable).

4.2 Defining the Simulation Model

For the purposes of our study we adopted the simplified model proposed by [10] and further described in [5]: the behavior of the animal species consists of the actions moving, feeding, reproducing and dying; an individual is represented as an n-tuple of quantifiable attributes. The fish class is the simplest one modeled as the 2-tuple

$$(C_{offs},\ A_{repr}),$$

consisting of the reproduction maturity age and the number of offspring created, which are positive natural numbers. Shark and whale objects are further identified by their life energy (a positive real number), that is constantly depleted and can be replenished only by killing prey, and by the energy gain from eating the prey (a positive natural number), which varies according to the prey type. Unlike fish, predators do not reproduce over a constant period, but do so only if a certain life energy level has been reached, thus their reproduction rate is directly related to the presence of prey [42]. The shark class is modeled as the 4-tuple

$$(C_{offs},\ E_{repr},\ E_{life},\ E_{eat\_fish})$$

and the whale class as the 5-tuple

$$(C_{offs},\ E_{repr},\ E_{life},\ E_{eat\_fish},\ E_{eat\_shark}),$$

respectively. Each existing individual performs the aforementioned actions as defined in [5] in a specified order and a cycle in the simulation ends when all individuals have 'acted', after which the environment is updated. In terms of cellular automata the actions of an individual/cell can be seen as the state transition of that cell or/and one or more of its neighbors, whereas the various instantiations of the respective attribute tuples as the private information about the given individual/cell.

5 Performed Tests and Results

5.1 Model Accuracy

In the previous work on the topic [5] the simulation was run with different combinations of initial population sizes, repeating each combination multiple times to account for the not entirely deterministic behavior of the individuals. The obtained results did show similarities with real world population dynamics [7], e. g. predator overpopulation leads to the prey going extinct, vice versa with prey underpopulation and predators are more vulnerable than prey since they have to roam and search for food. Moreover, by plotting the change in the population sizes over the course of the simulation, it was shown that despite the simplicity of the model, successful initial population size combinations result in the system reaching an *equilibrium* state, and the populations exhibiting dynamics like those described by the Lotka-Volterra model and presented in [24].

5.2 Ecosystem's Capability to Self-Sustain

Since not every initial population size combination is favorable regarding the ecosystem's survival, and some combinations have only a partial success rate, a second series of experiments was aimed to a) prove that there is an integral not empty solution space (initial population size combinations) for which the given model of the simulated ecosystem can reach the so called equilibrium state; and 2) to define its boundaries. Equilibrium state, also called stability, has a field of science depended definition. According to [11] the term used in context of ecosystem modeling will be a property of ecosystem where all components are acting together i.e. all species will survive the ongoing experiment and nothing foreshadow his nearly collapse. To discover and define the equilibrium state system was run automatically over a long period of time with randomly generated initial population sizes from a specified domain for the three species. It was shown that the successful input combinations form a set in 3-dimensional space, which with enough runs would resemble an irregular *solid*. In other words a solution space exists and the ecosystem is with high probability capable to self-sustain given any input from the respective set.

5.3 Input/Output Relation

From the results discussed in 5.1 and 5.2 it is clear that the system has a high chance to reach equilibrium for some input combinations, and a very low chance for others, i. e. a relation between the input parameters and the outcome of the simulation presumably exists. This can be proven formally by finding a function $f(p_1, p_2, ..., p_n)$ that maps an instance of the input parameters $p_1, ..., p_n$ to the respective output. As already stated, however, the problem under consideration is in practice analytically intractable, thus such a function cannot be obtained with a reasonable amount of effort. Instead, in our approach we decided to reuse the *empirical* formula described in [5], which is based on the statistical data gained from the initial tests and on the observations made while analyzing it. The function is defined as $S(P_f[n_f], P_s[n_s], P_w[n_w])$, where S is the dependent variable (ecosystem stability), and $P_f[n_f], P_s[n_s], P_w[n_w]$ are the independent variables representing the probability with which the respective initial population sizes of fish, sharks and whales would survive. Moreover, from the already made conclusions it is known that some species are more vulnerable than others, and so their survival should be weighted more, e. g. sharks. One the other hand, as stated in [23], some may contribute relatively little to ecosystem properties. As a result the stability of the system is defined as a function of the survival probability of each species, given its initial size, combined with a weighting factor based on the species' vulnerability:

$$S(P_f[n_f], P_s[n_s], P_w[n_w]) = P_f[n_f] \cdot 0.2 + P_s[n_s] \cdot 0.5 + P_w[n_w] \cdot 0.3 \qquad (1)$$

where $P_x[n_x]$ is the survival probability of species x with initial size n_x in the respective test case. Using this approximation formula the ecosystem's stability factor was calculated for various initial predator population sizes within the range [2 , 50000] and two distinct fish population sizes: 5,000 and 50,000. The results are illustrated in Fig. 2 and show an obvious relation between

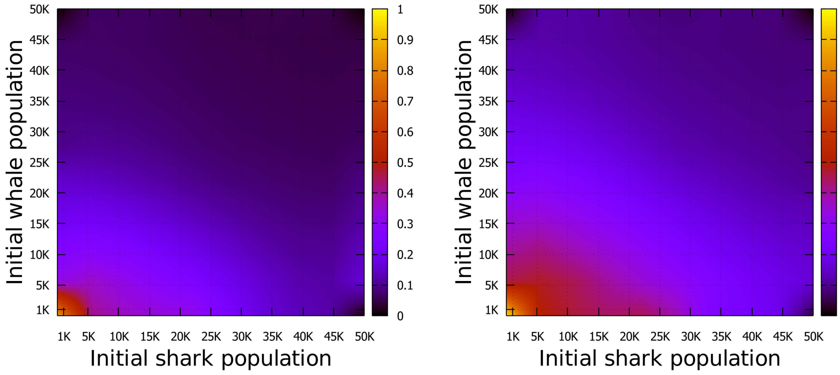

Fig. 2 Relation between the initial predator population sizes and the stability of the ecosystem obtained with Eq. 1. Two different fish population sizes have been used in the experiments: 5,000 (left) and 50,000 (right). Darker/colder colors represent lower stability factor. The stability factor ranges from 0.0 (no species survived any of the test cases) to 1.0 (every species survived all of the test cases).

the starting conditions and the simulation outcome. It is easy to see that an increasing predator population decreases the stability factor rapidly, indicated by the colors becoming darker towards the maximum values of the x- and y-axis. This is especially true for the whale population since the heat map darkens faster vertically (along the y-axis), rather than horizontally (along the x-axis). Ample quantities of fish prey also contribute for a better ecosystem stability (compare the visibly brighter heat map for the test cases involving an initial fish population of 50,000).

These results correspond to the conclusions made earlier about the accuracy of the model (see section 5.1). Any combination representing a predator overpopulation resp. prey under-population is colored in dark. Additionally, the greater part of both heat maps is in blue/purple hue, indicating a value less than 0.3. In other words, in most test cases both predator species, but especially the sharks, went extinct before the stopping condition of the simulation was met. Nevertheless, a comparison with the results regarding the system's ability to self-sustain documented in [5] reveals a partial error in the solution space's bounds. It is our belief that this could be eliminated with a better approximation of the solution space boundaries (by performing more runs) as suggested in the corresponding section or a light modification of the weighting factors used in Eq. 1.

Other set of tests we conducted were run to determine a similar relation between the system's stability factor and the initial numerical values of a species' attribute tuple. For that purpose we repeated all test cases illustrated in Fig. 2 (left), while only changing the instance of the fish species' attribute tuple from (2, 20) to (5, 20), i. e. increasing the maximum number of offspring a fish can

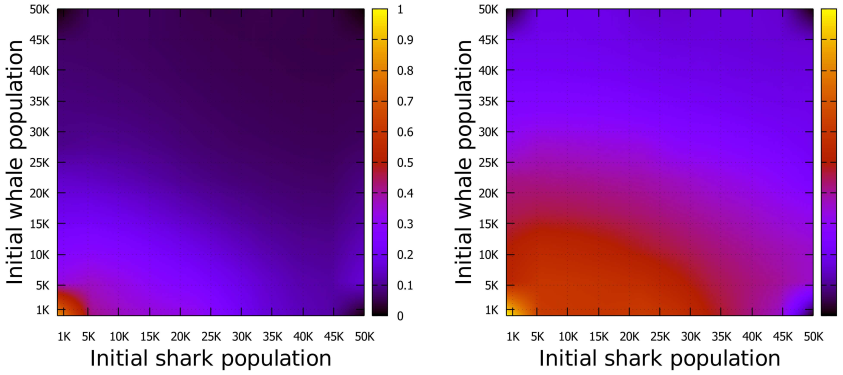

Fig. 3 Relation between the attribute 'offspring count' for the fish species and the stability of the ecosystem obtained with Eq. 1. Two different instances of the maximum offspring count have been used in the experiments: 2 (left) and 5 (right).

spawn from 2 to 5. This yielded a definite increase in the ecosystem's chance to reach equilibrium, as highlighted by the considerably brighter coloring of the heat map even in predator overpopulation/prey under-population scenarios (see Fig. 3, right).

The stability information gathered from all tests listed in Table 1 and from additional ones is visualized in Fig. 4 (left) as a function of the predator initial populations. It is easy to see that an increasing whale population is decreasing the stability rapidly (indicated by the sharp steps along the direction of the whale axis), whereas that of the sharks in a more gradual manner (indicted by the smaller steps along the shark axis). Beyond some point, however, the joint number of predators becomes too large for the environment to sustain.

A number of additional tests using random initial populations between 10 and 50,000 were conducted without repetition to gain insight regarding the boundaries of the set of successful combinations. Out of several hundred tests only those were chosen, where the system reached a balanced state. This subset of the tried input combinations can be seen in Fig. 4 (right). It is apparent that the fish population's size can vary, whereas that of the whales must not exceed 10,000. Similarly, the upper limit for the sharks is around 25,000. These results correspond to the information regarding the system's stability shown in Fig. 4 (left), and therefore support the approximated coefficients used in (1). To further examine the validity of the results one more test was made with initial populations taken from the subset illustrated in Fig. 4 (right). Since the subset is just an approximation of the actual one, moderate values were used taken from its center rather than its edges: 40,000 fish; 3,500 sharks: 4,500 whales; The test yielded a success rate of 200 out of 200 repetitions.

Unfortunately, the data shown so far is not very informative regarding the dynamics of the ecosystem, i.e. how do the species' populations vary over

Table 1 The effect of the three species' initial population size on the environment's stability calculated with (1).

	Initial Population			**Survival Probability** $P(x)$			
#	**Fish**	**Sharks**	**Whales**	**Fish**	**Sharks**	**Whales**	**System Stability**
(1)	10000	50000	10000	0.0	0.0	0.0	0.0
(2)	10000	50000	50000	0.0	0.0	0.0	0.0
(3)	50000	50000	50000	0.0	0.0	0.0	0.0
(4)	50000	50000	10000	0.1	0.0	0.0	0.02
(5)	100	10000	10000	0.6	0.0	0.0	0.12
(6)	10	10	10	1.0	0.0	0.0	0.2
(7)	10	100	100	1.0	0.0	0.0	0.2
(8)	100	10	10	1.0	0.1	0.1	0.28
(9)	1000	100	10000	1.0	0.0	0.3	0.29
(10)	100	1000	10000	0.9	0.0	0.4	0.3
(11)	100	10	100	1.0	0.0	0.5	0.35
(12)	1000	10000	10000	1.0	0.0	0.6	0.38
(13)	100	100	100	1.0	0.1	0.55	0.415
(14)	1000	1000	10000	1.0	0.0	0.8	0.44
(15)	100	10000	100	1.0	0.0	0.9	0.47
(16)	100	100	1000	1.0	0.0	1.0	0.5
(17)	100	1000	100	1.0	0.6	0.1	0.53
(18)	100	10000	1000	1.0	0.1	1.0	0.55
(19)	100	1000	1000	1.0	0.2	1.0	0.6
(20)	50000	10000	10000	1.0	0.3	1.0	0.65
(21)	10000	1000	10000	1.0	0.4	1.0	0.7
(22)	10000	1000	1000	1.0	0.8	1.0	0.9
(23)	1000	100	100	1.0	1.0	1.0	1.0
(24)	1000	1000	100	1.0	1.0	1.0	1.0
(25)	1000	10000	1000	1.0	1.0	1.0	1.0

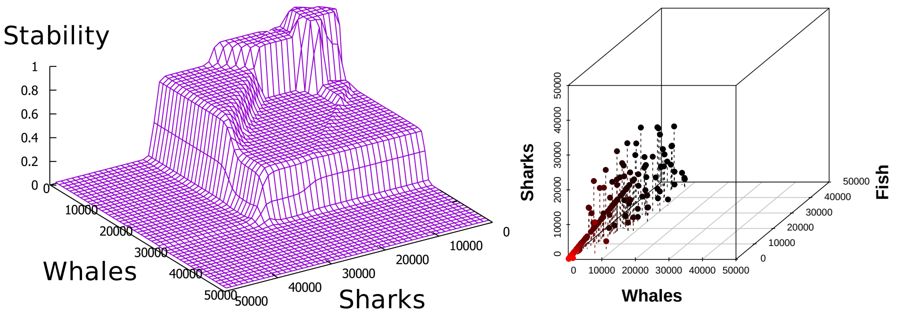

Fig. 4 The stability of the system for various initial predator populations calculated with (1) (left) and a set of initial populations for all three species, with which the system reaches a balanced state (right).

the course of the simulation. Therefore, the changing population sizes were recorded as a function of the time. A plot produced from such data can be seen in Fig. 5. Despite the simplicity of the model, one can easily distinguish the famous predator-prey relation described by the Lotka-Volterra model [24]. With the increase of the prey population the predators thrive and increase in

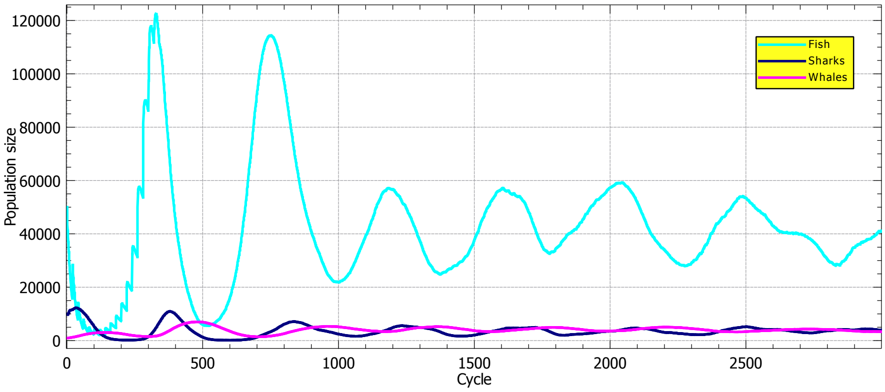

Fig. 5 A graph plotting the population sizes changing over the course of the simulation as a function of the time. The input consisted of 50000 fish (cyan), 10000 sharks (dark blue) and 1000 whales (pink).

numbers as well. The latter coupled with environmental factors such as overpopulation or diseases lead to a decrease in the prey population. Accordingly, soon after the predators dwindle as food becomes sparse. If the populations are big enough and no devastating disaster happens, the system will oscillate between some relatively constant boundaries, i.e. be stable. Past the 1000th cycle a similar phenomenon can be seen in Fig. 5. All three populations increase and decrease periodically with that of the sharks being slightly delayed relative to the fish population, and that of the whales relative to both fish and shark.

5.4 Approach to Efficiently Optimize Input Parameters

The final question we would consider during this work is how to optimize the entire set of input parameters, so that the outcome of the simulation is favorable. Up to now only the size of the initial populations was regarded, but when comparing the improvement of the system's stability by changing these as depicted on Fig. 2 with the improvement gained by modifying the attribute tuple of a species (see Fig. 3), one can quickly acknowledge the significantly better results of the latter approach. A major difficulty, however, is again the size of the input space. By using the 3-tuple $(Fish,\ Sharks,\ Whales)$ as input, where each element represents the respective species' population size within the range [10 , 50000], yields a total of

$$44991^3 \approx 50000^3 = 1.25 \cdot 10^{14}$$

possible input combinations. In a similar way we can count the possible instances I for the attribute n-tuple of each species with Eq. 2:

$$I = \prod_{i=1}^{n} |A_i|, \tag{2}$$

where $|A_i|$ is the number of distinct numerical values, that attribute A_i can take. For instance the 2-tuple of the fish species could have the following ranges for its elements

$$(C_{offs},\ A_{reproduce}) \Rightarrow ([1\ ,\ 5],\ [1\ ,\ 100]),$$

yielding a total of $I_{fish} = 5 \cdot 100 = 500$ possible attribute instances. Analogously the shark 4-tuple and the whale 5-tuple could be defined as

$$(C_{offs},\ E_{repr},\ E_{life},\ E_{eat\_fish}) \Rightarrow ([1\ ,\ 5],\ [51\ ,\ 100],\ [1\ ,\ 100],\ [1\ ,\ 10])$$

and

$$\begin{gathered}(C_{offs},\ E_{repr},\ E_{life},\ E_{eat\_fish},\ E_{eat\_shark}) \Rightarrow \\ ([1\ ,\ 5],\ \ [101\ ,\ 200],\ [1\ ,\ 200],\ [1\ ,\ 10],\ [1\ ,\ 50]),\end{gathered}$$

yielding $I_{shark} = 5 \cdot 50 \cdot 100 \cdot 10 = 250000$ and $I_{whale} = 5 \cdot 100 \cdot 200 \cdot 10 \cdot 50 = 50000000$ possible instances respectively. The total count of input parameter instances I_{total} is then

$$I_{total} = I_{fish} \cdot I_{shark} \cdot I_{whale} = 500 \cdot 250000 \cdot 50000000 = 6.25 \cdot 10^{15}\ .$$

Even though this number is not much greater than the number of population size combinations, there are 11 input parameters when summing the tuple elements of all three species, yielding an 11-dimensional input space respectively, unlike the 3-dimensional one for the population sizes. Therefore, applying the graphical solution approaches presented so far in the previous sections would be less than feasible. Moreover, the conclusions made were obtained using excessive testing at the cost of hundreds of hours of computational processing, and could be classified as good approximations at best, which is supported by the small contradiction between the results described in sections 5.2 and 5.3. This raises doubts regarding the efficiency of the simulation-based problem solving should problem complexity increase (e. g. more input parameters).

A more promising approach, that we would like to propose, is to combine the simulation-driven architecture of the cellular automaton as it is with an evolutionary search algorithm to traverse the input space for combinations that yield a stable ecosystem. Essentially, we would transform a simulation problem into an *optimization* one, where a model consisting of objects with specific characteristics and rules for their interaction is known, a desired output range is specified and the point of interest is the unknown set of input instances that yield an output within the desired range.

Inspired by biological evolution, hence their name, *evolutionary algorithms* (EA) have gained significant importance in the recent years, especially in the domain of optimization. Their use can be quite rewarding as an efficient alternative to traditional deterministic methods, more so in very complex and intractable problem scenarios. In the following a short summary of the basics is given, as well as an outline of the modifications needed to incorporate the evolutionary optimization technique in our application. It is our strong belief that the integration effort would not be considerable, but the obtained insights might prove to be of value. For a more comprehensive discussion on the topic of EA basics and application refer to [21,35,13].

*Solution candidate space.*As the name implies this is the set of all possible candidate solutions. It can be indefinitely large, but for practical reasons it has to be bounded. Candidate solutions can be anything from numerical values to entire objects. In the case of our Wa-Tor application, the candidate solution space consists of all attribute combinations for the three species (e. g. $6.25 \cdot 10^{15}$ if the aforementioned example is taken).

*Genetic operators.*These operators are functions that randomly modify the existing collection of candidate solutions to increase diversity. Typical operators are a) *mutation* — changing a random characteristic of a candidate solution; and b) *recombination* — combining two or more solutions to produce new ones; Integrating this functionality in the existing Wa-Tor simulation can be done quite easily in the reproduction action of the animals, e. g. when offspring are spawned their attribute instance can be created by combining that of the parent with those of other existing individuals. Mutation can be implemented similarly as a random change in a spawned offspring's attribute instance inherited by the parent. Genetic operators modify only the genotype. The phenotype cannot be inherited, but is crucial for the fitness of the respective candidate.

*Solution candidate encoding.*A candidate solution is typically represented as a *phenotype-genotype* pair. Just like in biology the phenotype is comprised of a candidate's visible attributes that change over its lifespan due to interaction or external events. On the other hand the genotype is the entire information about the core structure of a candidate in some encoded form, e. g. DNA in real life and usually a binary string in software applications. In the Wa-Tor scenario the phenotype of an object would be its actual instance in the simulation, whereas an adequate genotype would be the information about the object's attributes received at its creation, that is immutable during the lifespan of the object, but inheritable by its offspring.

*Fitness function.*The idea behind the fitness function is to decrease the population diversity by eliminating poor candidate solutions, but thus improve the overall quality of the candidate solution space (called 'survival of the fittest' or ' natural selection' in the field of biology). The fitness function in the Wa-Tor scenario is the simulated interaction between the individuals. Using the predefined interaction rules such as moving and feeding, individuals may survive or get killed by predators/die out of starvation. Certain genotype instances may have higher chances to prevail than others, e. g. individuals that can give birth to multiple offspring are more likely to pass down and spread their genotype in the respective population.

6 Conclusion and Future Work

The main goal of our project was to design and develop an application to test the strengths of a simulation-based analysis. A relatively simple and well-known problem domain was chosen, which, however, poses multiple challenges regarding its formal specification, hence the need of an alternative approach. An ecosystem and various species inhabiting it were modeled in the fashion of Dewdney's Wa-Tor [10]. Even though the created model is a fairly basic one, including only several interaction rules between the organisms within the simulation, and a handful of attributes to characterize them, no mathematical specification of the system can be derived without a significant amount of effort. Via simulation however, it was possible to define very simple rules on a micro level, and still obtain valuable insights regarding the global behavior of the environment and the populations inhabiting it.

As a preliminary work on the topic we performed extensive testing to gather enough data and visualize the relation between the inputs and the output of the system. For that an empirically derived formula (see Eq. 1) was used to calculate the system's output (stability factor) as a function of the probability that a species with a given population size would survive. The formula uses weight coefficients to discriminate between more robust and more vulnerable species. In Figures 2 and 3 we showed the direct dependence of the ecosystem's stability on the initial population sizes and on a specific species' attribute — the number of created offspring after reproduction. In the final part of the paper the challenges of extensive testing were discussed and a new approach based on evolutionary programming was proposed as a possible more efficient alternative.

Future work would revolve around the expansion of the application to incorporate the described evolutionary technique and allow for the model (species' attributes) to change over the course of the simulation similar to the evolution of organisms in the real world. The potential optimization benefits of this approach are to be evaluated and compared to those of the extensive testing. Moreover, new typicality- and eccentricity-based data analytics method (TEDA) presented in [3] might be applied to detect anomalies within cellular space. Anomalies detected by TEDA can provide information on the stability of small centres of individual species and their influence on the global population sustainability. By introducing a generic framework of the knowledge and data integration (KDI) and paradigms of evolving fuzzy and neuro-fuzzy models, that means an evolving computational intelligence system (ECIS) could bring more perspectives in this approach [1]. Similarly, applying the CEDA method described in [18] can provide information about the species formed into so called *arbitrarily shaped* clusters thus demonstrating the population dynamics in smaller communities. An essential element of such an analysis is the ability to process the data in an on-line manner. Incoming data evolve over time (individuals disappear or their attributes are changing) therefore forming not-regular shaped clusters. Spatial analysis of these clusters can lead to conclusions about population self-sustaining and ability to achieve a state of equilibrium.

We plan to apply presented simulation framework for further real-life examples. Modeling the honey bee population from [27] show that population size is related with forager bees death rate. Quantitative modeling results show that if death rate is sustained higher than some threshold colony failure is inevitable. This thesis could be set against the cellular automata system like it was done in [26]. In such research species should be now defined as forager bees and their enemies (bee-eater birds, wasps, temperature dropout or unexpected heavy rain). Proper formulas and rules of interactions should be defined [17]. The Wa-Tor model should be extended and significantly expanded (e.g global number of bees factor should be included). Hive-specific threats might be also examined. Bee swarm simulation connected with presence of the enemies like parasite *Varroa destructor* could be an appealing issue. Varroa strikes only against younger bees what may lead to bees distinction in the future. Moreover, in [9] it was shown that bees change their mood depends on presence of a queen bee liable for laying eggs. Healthy queen is prerequisite for sustaining the colony life and any disturbances could lead to replacing disabled queen bee by newborn queen. Modeling the feelings of bees on the principles described in [32] in combination with the aging queen and thus laying less eggs (ever smaller population growth) will allow us to answer the question about the optimal amount of bees feeders, foragers and honey or weather dependencies.

References

1. Angelov, P., Kasabov, N.: Evolving computational intelligence systems. In: Proc. 1st International Workshop on Genetic Fuzzy Systems, pp. 76–82, (2005)
2. Angelov, P., Kasabov, N.: Evolving intelligent systems, eIS. In: IEEE SMC eNewsLetter, 15, pp. 1–13, (2006)
3. Angelov, P.: Outside the box: an alternative data analytics framework. In: Journal of Automation Mobile Robotics and Intelligent Systems, 8(2), pp. 29–35, (2014)
4. Bădică, A., Bădică, C., Ivanović, M., & Dănciulescu, D.: Multi-agent modelling and simulation of graphbased predatorprey dynamic systems: A BDI approach. In: Expert Systems, 35(5), exsy12263. (2018)
5. Balabanov, K., Fietz, R. G., Logofătu, D.: Considerations in Analyzing Ecological Dependent Populations in a Changing Environment. In: Computational Collective Intelligence, pp. 223–232, vol 10448, Part I (September), Nicosia (2017)
6. Balabanov K., Logofătu D., Badica C., Leon F., A Simulation-Based Analysis of Interdependent Populations in a Dynamic Ecological Environment. In: Iliadis L., Maglogiannis I., Plagianakos V. (eds) Artificial Intelligence Applications and Innovations. AIAI 2018. IFIP Advances in Information and Communication Technology, vol 519, pp. 437–448, Springer, Cham (2018)
7. Begon, M., Mortimer, M., Thompson, D. J.: Population Ecology: A Unified Study of Animals and Plants, 3rd ed, [Online]: Wiley-Blackwell (1996)
8. Canyameres S., Logofătu D.: Platform for Simulation and Improvement of Swarm Behavior in Changing Environments. In: Iliadis L., Maglogiannis I., Papadopoulos H. (eds) Artificial Intelligence Applications and Innovations. AIAI 2014. IFIP Advances in Information and Communication Technology, vol 436, pp. 121–129, Springer, Berlin, Heidelberg (2014)
9. Cejrowski, T., Szymaski, J., Mora, H., Gil, D.: Detection of the Bee Queen Presence Using Sound Analysis, Asian Conference on Intelligent Information and Database Systems, pp. 297–306, Springer, Cham (2018)
10. Dewdney, A. K.: Sharks and fish Wage an ecological War on the toroidal planet Wa-Tor, Scientific American, pp. 14–22, 251 (December), (1984)
11. DiStefano III, J.: Dynamic Systems Biology Modeling and Simulation. 1st ed., Academic Press, Cambridge (January), pp. 21–23, (2015)

12. Dougoud, M., Vinckenbosch, L., Rohr, R. P., Bersier, L.-F., Mazza, C.: The feasibility of equilibria in large ecosystems: A primary but neglected concept in the complexity–stability debate, PLoS Comput Biol 14(2), pp. 1–18, (2018)
13. Eiben, A. E., Smith, J. E.: Introduction to evolutionary computing. 2nd ed. Springer, Heidelberg (2015)
14. F-Droid, Wa-Tor - A simple population dynamics simulator, mobile application, `https://f-droid.org/en/packages/com.dirkgassen.wator`, April, (2016), last accessed: 06.11.2018
15. Farge, M.: Numerical experimentation: A third way to study nature. In: Frontiers of Computational Science. Proceedings of the International Symposium on Frontiers of Computational Science 2005, pp. 15–30. Springer-Verlag, Berlin (2007)
16. Floreano, D., Mattiussi, C.: Bio-inspired artificial intelligence : theories, methods, and technologies. MIT Press, Cambridge (2008)
17. Hogeweg, P., Hesper, B.: The ontogeny of the interaction structure in bumble bee colonies: a MIRROR model, Behavioral Ecology and Sociobiology, 12(4), pp. 271–283, (1983)
18. Hyde, R., Angelov, P., MacKenzie, A. R.: Fully online clustering of evolving data streams into arbitrarily shaped clusters. Information Sciences, 382, pp. 96–114, (2017)
19. Gardner, M: The fantastic combinations of John Conway's new solitaire game "life". In: Scientific American, pp. 120–123, 223 (April), (1970)
20. Gardner, M.: On cellular automata, self-reproduction, the Garden of Eden and the game "life". In: Scientific American, pp. 112–117, 224 (February), (1971)
21. Gerdes, I., Klawonn, F., Kruse, R.: Evolutionäre Algorithmen : genetische Algorithmen - Strategien und Optimierungsverfahren - Beispielanwendungen. Vieweg, Wiesbaden (2004)
22. Harold, F. M.: The Way of the Cell. Oxford University Press, Oxford (2001)
23. Hooper, D. U., Chapin III, F. S., Ewel, J. J., Hector, A., Inchausti., P., Lavorel, S., Lawton, J. H., Lodge, D. M., Loreau, M., Naeem, S., Schmid, B., Setälä, H., Symstad, A. J., Vandermeer, J., Wardle, D. A.: Effects of Biodiversity on Ecosystem Functioning: a Consensus of Current Knowledge In: Ecological Monographs, pp. 3–35 (February), (2005)
24. Hoppensteadt, F.: Predator-prey model. In: Scholarpedia, pp. 1563, 1 (October), (2006)
25. Huang, Z., D. V. Rosowsky, P. R. Sparks, Hurricane simulation techniques for the evaluation of wind-speeds and expected insurance losses, Journal of wind engineering and industrial aerodynamics 89.7-8, pp. 605-617, (2001)
26. Jafelice, R. M., da Silva, P. N.: Studies on Population Dynamics Using Cellular Automata, In Cellular Automata-Simplicity Behind Complexity. InTech., (2011)
27. Khoury, D. S., Myerscough, M. R., Barron, A. B.: A Quantitative Model of Honey Bee Colony Population Dynamics. In PLoS ONE 6(4): e18491. `doi:10.1371/journal.pone.0018491` (2011), last accessed: 06.11.2018
28. Law, A. M.: Simulation modeling and analysis. McGraw-Hill Higher Education, New York (1997)
29. Logofătu, D., Sobol, G., Stamate, D., Balabanov, K.: A Novel Space Filling based Approach to PSO Algorithms for Autonomous Agents. In: Computational Collective Intelligence, pp. 361–370, vol 10448, Part I (September), Nicosia (2017)
30. Logofătu, D., Sobol, G., Andersson, C., Stamate, D., Balabanov, K., Cejrowski, T., Particle swarm optimization algorithms for autonomous robots with deterministic leaders using space filling movements. In: Evolving Systems, pp. 1–14 (2018)
31. Mallet, D. G., De Pillis, L. G.: A cellular automata model of tumorimmune system interactions. In Journal of Theoretical Biology, pp. 334–350, vol 239, Issue 3, April, (2006)
32. Masuch, M., Hartman, K., Schuster, G.: Emotional agents for interactive environments. In Creating, Connecting and Collaborating through Computing. Fourth IEEE International Conference on Creating, Connecting and Collaborating through Computing (C5'06), pp. 96–102, January, Berkeley (2006)
33. Max Strauch, Modeling and Simulation, Cute simulations, `https://maxstrauch.github.io/projects/mod-sim/index.html`, last accessed: 06.11.2018
34. McCann, K. S.: The diversity-stability debate, Nature, vol 405, pp. 228-233 (2000)
35. Michalewicz, Z.: Genetic algorithms + data structures := evolution programs., 3rd ed., Springer, Berlin (2008)
36. Norris, J. S.: Mission-critical development with open source software: lessons learned. In: IEEE Software, pp. 42–49, 21 (January), (2004)

37. Power, D. A., Watson, R. A., Szathmáry, E., Mills, R., Powers, S. T., Doncaster, C. P., Czapp, B.: What can ecosystems learn? Expanding evolutionary ecology with learning theory, Biology Direct, pp. 1–24, (2015)
38. Reynolds, C. W.: Flocks, herds and schools: A distributed behavioral model. In ACM SIGGRAPH computer graphics, vol 21, No. 4 (August), pp. 25–34, (1987)
39. Rédei, M.: John von Neumann: Selected letters. RI: American Mathematical Society, Providence (2005)
40. Southwood, T. R. E., Henderson, P. A.: Ecological methods. Wiley-Blackwell, 4th ed., pp. 29–31, (2016)
41. Tilman, D., Reich, P. B., Knops, J. M. H.: Biodiversity and ecosystem stability in a decade-long grassland experiment, Nature, vol 441, pp. 629–632 (2006)
42. Tschirhart, J: General Equilibrium of an Ecosystem, Journal of Theoretical Biology, pp. 1–41, (2000)
43. Ulam, S. M.: Adventures of a Mathematician. Scribner, New York (1976)