

Simulation of an ecosystem on a toroidal lattice with three species

Paolina Gadjulova

1 Introduction

In 1984, Alexander Dewdney introduced the dynamic cell process called “Wa-Tor” in which two species – sharks and fish, inhabit a rectangle cell table with the properties of a toroidal lattice. Each of the organisms must move stochastically in one of its neighbour cells on a defined unit of time, called a chronon. The prey moves first and the predator moves second. Each of the organisms reproduces on a defined period of time. The sharks die of starvation if they have not eaten for a certain number of chronons. The posed question is the following – are there such starting parameters with which the ecosystem would stay in balance. An ecosystem is assumed to be balanced if at neither of the species ever completely disappear [1]. The practical applications of the simulation are shown in a parallel computing paper by the *California Institute of Technology* and *Jet Propulsion Laboratory* [3] and later the process is examined from both a mathematical and an ecological point of view [4].

Dewdney has showed example parameters for an ecosystem with two species. In the current paper, we consider a question posed by David Keyes in 2014 – is it possible to have a balanced ecosystem with three species, forming a food chain. In addition, we examine whether it is possible to find boundary points for the starting parameters. In a previous work, taken place on the Saudi Research Science Institute, one has shown examples of an ecosystem with three species surviving the first 1,500 chronons of the simulation before the middle species completely disappeared [2].

Balancing an ecosystem like this one turns out to be a very delicate process. The population sizes of the species are extremely dependent on the stochastic starting conditions (the periods of reproduction and starvation). The predator-prey model of Lotka-Volterra and its different generalizations describe similar dynamic processes in the classic modelling.

In the current paper, we use the classic metaheuristic, genetic algorithms to find suitable starting conditions. The method is using an evolution analogy for searching a suitable solution to an optimization problem – each possible solution is represented as an individual and one generates a new population on every iteration of the algorithm. New populations are obtained by crossing two parental individuals, from the previous population. Using this technique we have reached starting conditions providing the wanted cyclic behaviour of the system.

In the next section, the problem is formally defined and we describe both classic predator-prey models and genetic algorithms. In section 3 we show our genetic algorithm

and in section 4 – the results obtained by it. In section 5 we study the cyclic behaviour of the system in two ways – in section 5.1 via autocorrelation between the sizes of population and in section 5.2 we modify the ecosystem to a more simple memoryless model. The modification has some relations to mutually catalytic branching random walks. In section 6 we have suggested an algorithm for finding boundary points of the population sizes when the other parameters are fixed.

The full version of the paper and additional materials as well as implementations of the codes are available at https://github.com/polly013/wator_project.

2 Methods and definitions

2.1 The problem

Three species inhabit a toroidal table (the distances between the left side and the right one, and between the bottom one and the top one, are one unit). The table is with the size of $N \times M$. Each cell can contain either one of the species or nothing. The species form a food chain and for convenience in the paper we will name them plankton, fish and sharks. The sharks are on the top of the chain and they are feeding only on fish, the fish are feeding on the plankton and the plankton is the producer of the chain. Each of the organisms in the simulation moves to one of its neighbour by side cells on a defined unit of time, called a chronon. If there is no uninhabited neighbour cell, the current organism does not move. The movement is performed by the order in the food chain – the plankton moves first, then the fish and after that are the sharks. Reproduction periods are defined for each of the species. The process is done along with the movement of the organisms – if the current organism must reproduce, the new one is placed on the previous cell. Similarly, if all of the adjacent cells are inhabited, then the organism does not reproduce and the operation is dismissed until the next period of reproduction. Also, starvation periods are defined for predator species – we assume that the bottom species in the food chain is the producer.

We will refer to the following eight parameters as starting conditions – population sizes of the species in the first chronon of the simulation, reproduction periods for each of the species and starvation periods for the predator species. The aim of the current research is to find suitable starting conditions with which the ecosystem would stay in balance for undefined period of time regardless of the distribution of the organisms on the torus.

2.2 Lotka-Volterra models

Lotka-Volterra equations are a simple non-linear, differential system usually used to describe the interactions between predator-prey species. Let $x(t)$ denotes the prey population number at time t and $y(t)$ denotes the predator one. In the original equations the populations vary in time in the following way:

$$\begin{aligned}\dot{x} &= x(a - by) \\ \dot{y} &= y(-c + dx),\end{aligned}$$

where $a, b, c, d, \geq 0$.

Here a and c regulate the behaviour of the species in the absence of the other one and b and d – the impact of each species on the other. Thus, it is assumed that the growth rate of the prey is a positive constant and this rate decreases linearly as a function of the predator population. Similarly, the growth rate of the predators in the absence of the prey is negative and increases linearly as a function of prey density.

There are many Lotka-Volterra generalizations and modifications, aiming to adapt the model to be more realistic. Competitive Lotka-Volterra equations are one of the most important modifications of the model. Here the competitiveness between the organisms from the same species are expressed by $-ex$ and $-fy$.

$$\dot{x} = x(a - ex - by)$$

$$\dot{y} = y(-c + dx - fy),$$

where $a, b, c, d, e, f \geq 0$.

Lotka-Volterra equations for three species in a food chain are another generalization, coherent with the problem exposed in this paper.

$$\dot{x}_1 = x_1(r_1 - a_{11}x_1 - a_{12}x_2)$$

$$\dot{x}_2 = x_2(-r_2 + a_{21}x_1 - a_{22}x_2 - a_{23}x_3)$$

$$\dot{x}_3 = x_3(-r_3 + a_{32}x_2 - a_{33}x_3),$$

where each r_i and a_{ij} are positive. Similarly to the original equations, r_i regulates the behaviour of each population and a_{ij} shows the interactions between i^{th} species and j^{th} one, with included competitiveness.

A classic example illustrating the interactions between two predator-prey species in the nature is the lynx and snowshoe hare data of Hudson Bay Company [1]. In *figure 1* we can see the populations of the species from 1850 to 1900 and in *figure 2* – the populations of the species according to Lotka-Volterra equations.

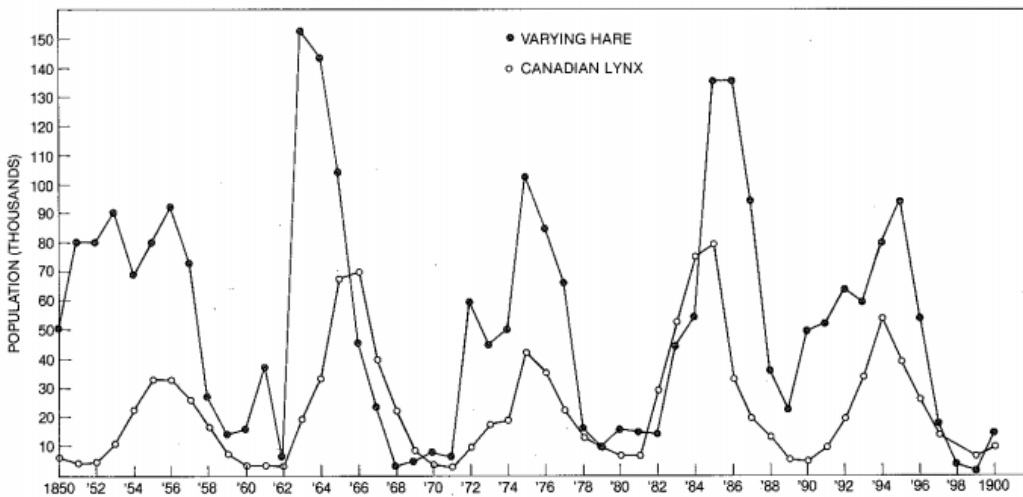


Figure 1: The populations of the lynx and the snowshoe hare, Hudson Bay Company's research [1]

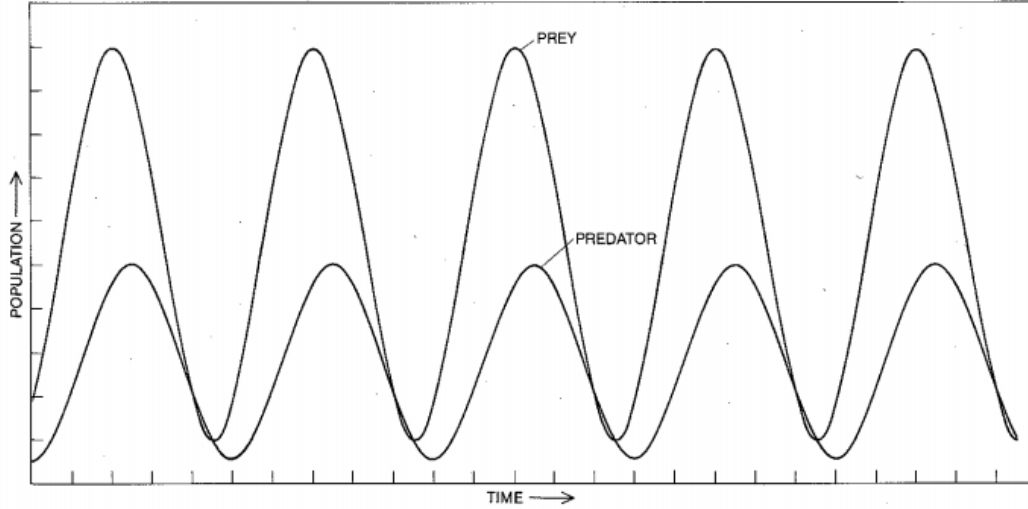


Figure 2: The populations of predator-prey species according to Lotka-Volterra equations [1]

2.3 Genetic algorithms

Definition 1. *Stochastic search* is a search method in which every next state depends on both the parameters of the process and a factor with a random behaviour.

Definition 2. *Beam search* is a heuristic search algorithm that explores a graph by expanding the most promising node in a limited set.

Definition 3. A *genetic algorithm* is a stochastic beam search in which the new states are generated by combining pairs of parent states.

Definition 4. An *individual* is called every candidate solution for the problem.

Definition 5. A *generation* is a set of individuals after a number of iterations of the algorithm.

Definition 6. A *population* is a set of the individuals in the same generation.

Definition 7. An *encoding* is a compact and convenient way of representing an individual.

Definition 8. A *chromosome* is an encoded individual with genes, where genes are the parameters of the solution.

Definition 9. Every individual has a *fitness function* which indicates how adjusted is the particular individual meaning how good the current solution is.

Definition 10. *Selection* is the process of choosing suitable parental individuals for reproduction.

Definition 11. *Crossing over* is the process of combining the genes of the two parent individuals.

Definition 12. A *mutation* is a change of a gene in a certain chromosome. This operation is usually performed with some probability.

The classical genetic algorithm looks in the following way:

1. Generate an initial generation. Usually this step is done randomly but it may be done by a heuristic method.
2. Every individual of the population is evaluated by the fitness function.
3. If the condition for terminating the algorithm is reached, the algorithm stops. Otherwise it continues.
4. Generate a new generation:
 - (a) **Elitism:** a part of the best individuals from the previous generation are carried over to the current, unaltered. This strategy is known as elitist selection and guarantees that the quality of the individuals, obtained by the genetic algorithm, will not decrease much from one generation to the next.
 - (b) **Reproduction:**
 - i. Two parent individuals are selected from the previous generation (by the selection method).
 - ii. The chosen individuals are crossed over.
 - iii. The new individual is mutated with some probability.
5. The algorithm is repeated from step 2.

3 Genetic algorithm searching for suitable parameters

Encoding Each chromosome contains the starting conditions as ordered eights – the population sizes of each species, the reproduction periods of each species and the starvation periods for the predator species.

Fitness function We consider the average surviving rate of the ecosystem, obtained by simulation of the process. The fitness function value is the mean of 20 runs of the system.

Selection We have implemented the well-known Roulette-Wheel selection. Using this method the probability to choose a particular individual is proportional to its evaluation given by the fitness function, which is one of the reasons the method is commonly used. First the sum S_i of the fitness functions for the current generation from the first individual to the i^{th} including is calculated. Afterwards, a random number r between 0 and S_N , where N is the size of the population is chosen. The candidate matching the condition $S_{i-1} \leq r < S_i$ is selected for reproduction. The probability for the i^{th} individual to be chosen is equal to $\frac{f_i}{\sum_{j=1}^N f_j}$.

Crossing over After the two individual for reproduction are selected, the new individual gene values are the mean of the parental ones.

Mutation A random gene in the current chromosome is chosen and its value is changed to a new valid one. This operation is performed with 5% probability.

Initial population The parameters in the initial population are chosen in a resembling the nature way. The bottom species in the food chain is the most numerous and its reproduction is the fastest. The second species (fish) is the next most numerous species, its reproduction is faster than the third species (sharks) and its starvation period is smaller than the third species starvation period.

In the implementation of the algorithm we have used the pseudorandom-number generator *Mersenne twister*.

4 Results obtained by the genetic algorithm

The ecosystem is implemented on a board with the size of 64×64 . In *figure 3* we observe the alteration of the population sizes of the species in the first 1,700 chronons of the simulation. In the example the starting parameters are 1094 plankton, 1074 fish and 91 sharks. They reproduce respectively on each 1, 11 and 11 chronons and starve on each 9 and 30 chronons. These parameters are obtained by the constructed by us genetic algorithm. The balance in the system remained even after 81,000,000 chronons in the simulation after the process was interrupted by us.

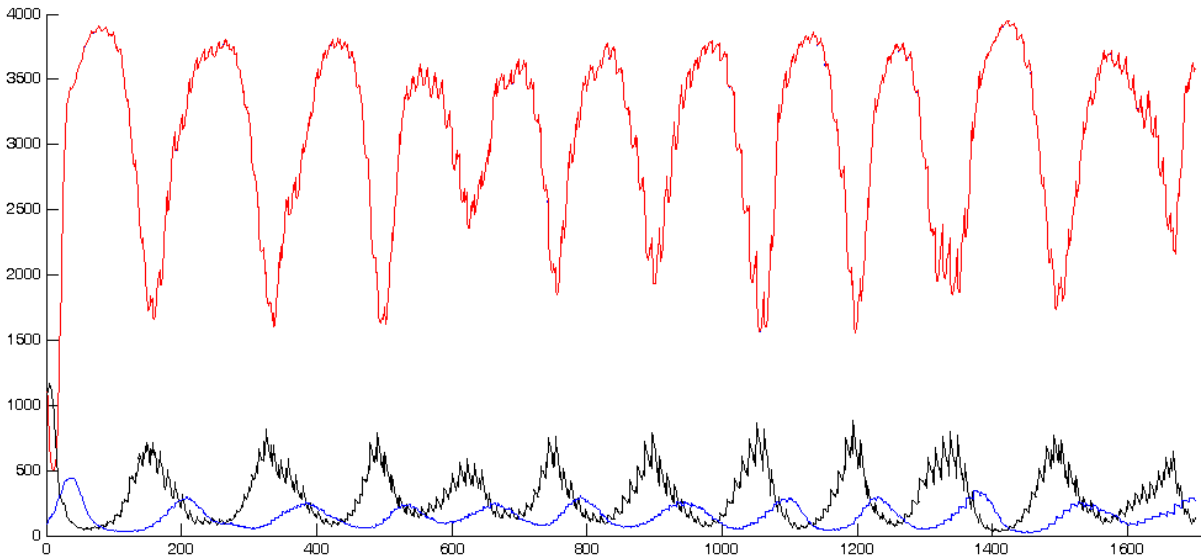


Figure 3: The population sizes in the first 1,700 chronons of the simulation. The plankton population is marked with red, the fish – with black and the sharks – with blue.

The data from the diagram resembles the classic Lotka-Volterra predator-prey model, which is presented in *figure 2*. A cyclic behaviour with shifting of the local extremums in the populations of the organisms is observed in both the examined ecosystem and the classic predator-prey model. The resemblance is distinctly expressed in a comparison between Lotka-Volterra phase portraits and the states of the simulation. The data of the simulation is displayed in the two following diagrams for ecosystem with two and three species.

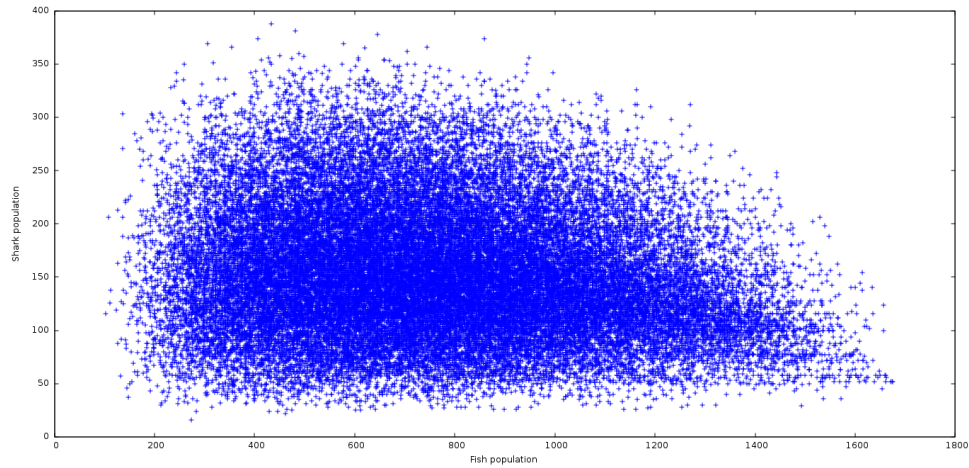


Figure 4: The diagram shows the states of the ecosystem with two species. The first 40,000 chronons of the simulation are displayed with blue.

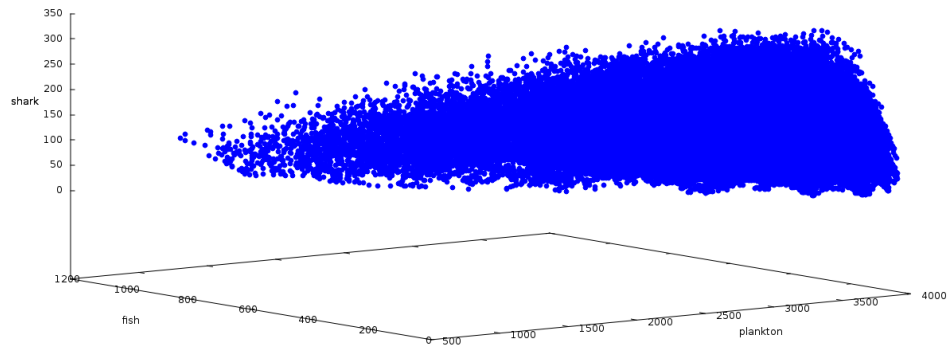


Figure 5: The diagram shows the states of the ecosystem with three species. The first 40,000 chronons of the simulation are displayed with blue.

In the next table are shown examples of parameters with which the system has cyclic behaviour. Again, these parameters are obtained by the genetic algorithm and they are examined to the first 100,000 chronons.

Table 1: Starting parameters, obtained by the genetic algorithm

P. num.	F. num.	S. num.	P. repr.	F. repr.	S. repr.	F. starv.	S. starv.
1094	1074	91	1	11	11	9	30
2124	802	61	1	4	9	6	19
672	597	231	1	15	24	2	27
938	860	138	4	17	26	11	29
1073	849	154	1	8	28	2	15
742	789	369	2	9	25	5	23
1156	1027	229	1	9	24	3	23
793	780	137	1	9	11	2	32
1264	1330	270	1	11	19	2	10
845	588	86	5	11	11	5	18
1191	845	206	1	20	24	8	35
1346	833	178	2	3	13	2	23
1018	1174	70	3	9	26	6	31

5 Study on the periodic behaviour of the system

5.1 Autocorrelation

In the current subsection we consider the autocorrelation for each species in the simulation.

Definition 13. The *autocorrelation* r for a period of time k with equally distributed in time measurement of the function Y is defined as [5]:

$$r_k = \frac{\sum_{i=1}^{N-k} (Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^N (Y_i - \bar{Y})^2}$$

The method is usually used in searching for repeatable events and has values between -1 and 1 . In the literature the time period k is usually referred as lag.

The autocorrelation between the population sizes of the species, measured after the first 100 chronons to the next 10,000, is showed in the following diagrams. The used starting parameters are from the last row in *table 1*. The data shows a significant autocorrelation levels. The smaller periods of cyclicity is related to the reproduction cycles of the species. The larger periods are relevant to the predator-prey behaviour. The levels of autocorrelation indicate that the process will behave cyclically and statistically improve the balance of the ecosystem.

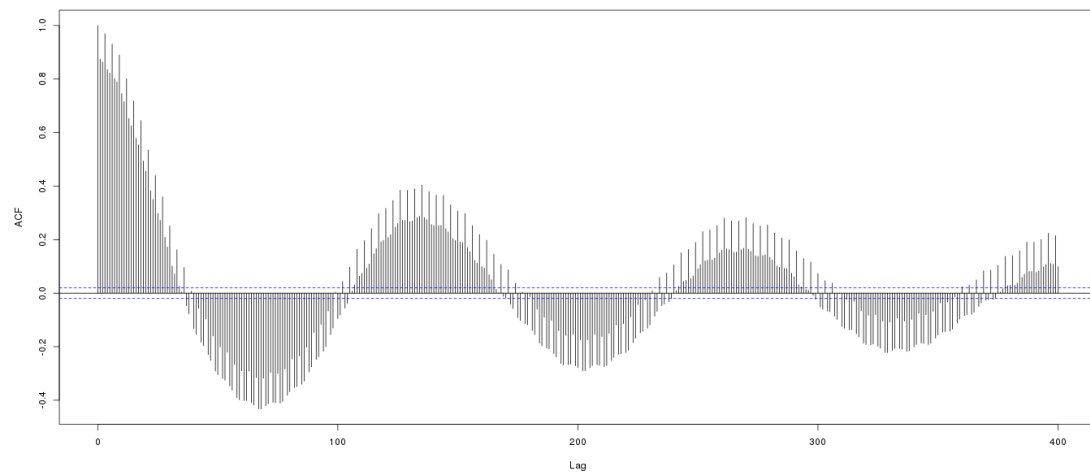


Figure 6: Autocorrelation for the plankton population

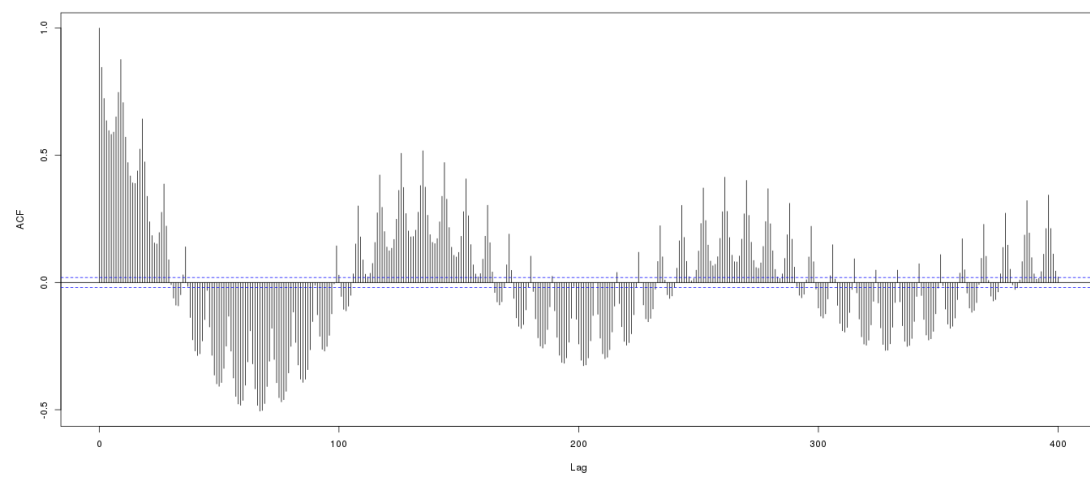


Figure 7: Autocorrelation for the fish population

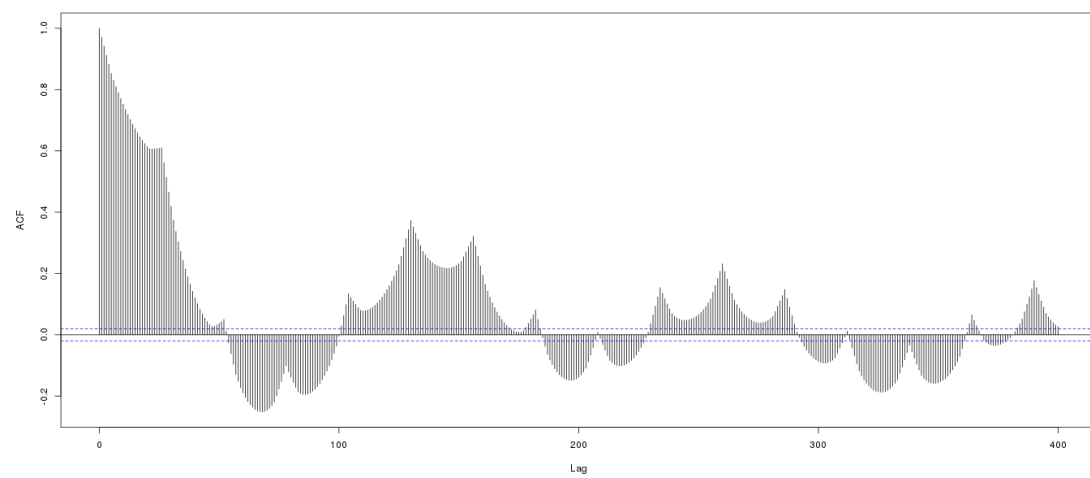


Figure 8: Autocorrelation for the shark population

5.2 Modification of the problem

We suggest a simplified model of the ecosystem, in order to ease the theoretical consideration of the problem. The modification is expressed in removing the need for memory – the organisms in the modification do not reproduce and starve on a certain period of time, but on each turn with a relevant probability ($\frac{1}{x}$, where x is the corresponding to the process parameter). In this way the current chronon depends only on the previous one. However, the modification is not trivial yet because the movement of the organisms while eating is not always stochastic.

We have tested the parameters from section 4 and the conclusion is that the balance is preserved with the exception of parameters having 50% probability to starve, in other words when the former parameter for starvation is 2.

The parameters with which the ecosystem stayed in balance for the first 100,000 chronons are showed in the following table.

Table 2: Starting parameters, obtained by the genetic algorithm

P. num.	F. num.	S. num.	P. repr.	F. repr.	S. repr.	F. starve.	S. starve.
1094	1074	91	1	11	11	9	30
2124	802	61	1	4	9	6	19
938	860	138	4	17	26	11	29
742	789	369	2	9	25	5	23
1156	1027	229	1	9	24	3	23
845	588	86	5	11	11	5	18
1191	845	206	1	20	24	8	35
1018	1174	70	3	9	26	6	31

6 Algorithm for finding boundary points for population sizes

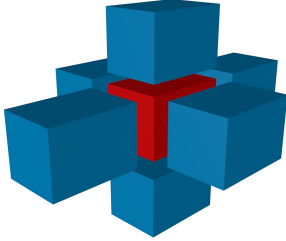
Let us fix the parameters for reproduction and starvation along with the sizes of the toroidal lattice and change only the parameters for the population sizes. In our study we have used the parameters from the bottom line in *table 1*. The reason why we have chosen them is that in some cases of the testing of the parameters the plankton disappeared in the first few chronons from the simulation which suggests that these parameters are close to the boundary between balanced and unbalanced states. When the plankton does not disappear the mean values for the states of the simulation are 3017, 296 and 155 respectively for the plankton, the fish, and the sharks. Let us represent each ordered triple of valid population sizes in the 3-dimensional space. The mean values of population sizes are on 3052 units distance from the initial triple $\{1018, 1174, 70\}$. Thus, if we have to consider each triple on a distance smaller than 3052 from the mean values, their number would be the number of the solutions in the following equation for m_1, m_2, m_3 and R .

$$|m_1 - n_1| + |m_2 - n_2| + |m_3 - n_3| = R$$

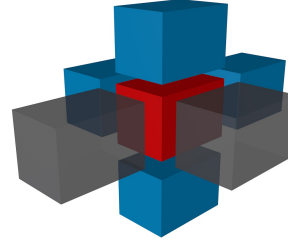
The solutions of this equation for $m_1 = 3107, m_2 = 296, m_3 = 155$ and $R = 3052$ are 12,053,832 and the sum of the solutions for which $0 \leq R \leq 3052$ are 13,554,816,987, making the computer exhaustion of each triples impossible. In the current research we suggest the following algorithm for examining a set of valid tripes.

Let us represent each triple of parameters as coordinates in the 3-dimensional space. We divide this space into identical parallelepipeds. We will consider only the center of the current parallelepiped and ignore the other triples in it. If the current triple is balanced after a finite period of chronons and defined runs of the simulation, the algorithm continues with the six adjacent parallelepipeds to the current one (its neighbour ones by side). If one of the species disappears, the algorithm continues only in a part of the adjacent parallelepipeds. It omits the following triples:

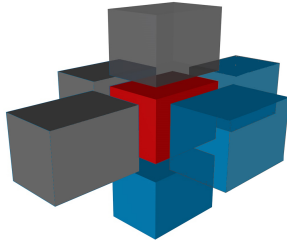
1. If in the triple (x, y, z) the first species disappears, it will also disappear in $(x-a, y, z)$ and $(x, y+a, z)$.
2. If in the triple (x, y, z) the second species disappears, it will also disappear in $(x-a, y, z)$, $(x, y-a, z)$ and $(x, y, z+a)$.
3. If in the triple (x, y, z) the third species disappears, it will also disappear in $(x, y-a, z)$ and $(x, y, z-a)$.



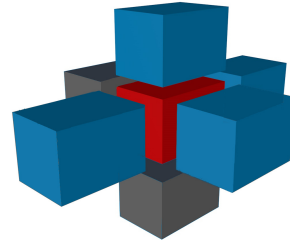
(a) The study will continue with the parallelepipeds in blue if the current one is balanced



(b) The study will continue with the parallelepipeds in blue if in the current one the plankton disappears



(c) The study will continue with the parallelepipeds in blue if in the current one the fish disappears.



(d) The study will continue with the parallelepipeds in blue if in the current one the sharks disappear.

The algorithm starts with a triple containing the mean values for the fixed parameters i.e. in our example the triple $\{3107, 296, 155\}$.

The current data is displayed in the following diagram. Considering the size of the data, the upper front is still incomplete and it is a subject of a further research.

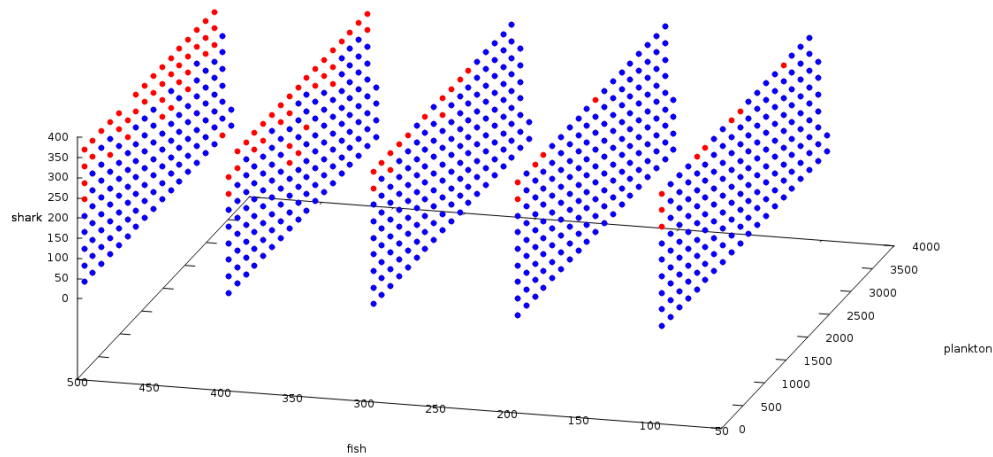


Figure 10: In blue are colored the pairs in which the ecosystem is stable for at least 100,000 chronons and in red – the pairs in which the ecosystem is stable for 100,000 chronons in half of the examples.

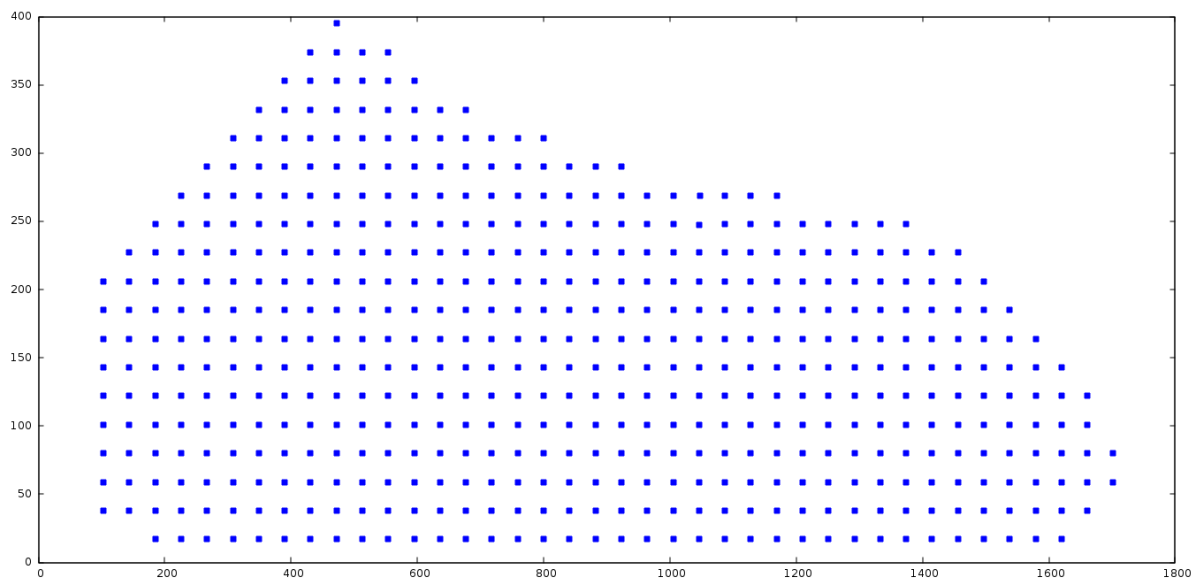


Figure 11: In blue are colored the pairs in which the ecosystem is stable for at least 100,000 chronons.

7 Conclusion and future development

In the current paper we have considered a similar system to the predator-prey model with three species. We have achieved a significant improvement of the previous results, using a genetic algorithm for finding suitable starting conditions and we have suggested a tool for studying the boundaries of the parameters. We have seen that the cell process has the same cyclic behaviour as the predator-prey model and also has a shifting of the local extremums, which is a typical feature of Lotka-Volterra equations. The considered ecosystem has some similarities to the mutually catalytic branching random walks, which is a suggestion to continue the study with theoretically proving the cyclic behaviour of the system. The research will continue with searching for a stable region for each of the parameters, with studying the relations between the parameters and with further comparisons with Lotka-Volterra equations. We will make a further research of the boundary points using the algorithm suggested in section 6 and we will examine the original problem with two species using Lotka-Volterra differential equations and providing example parameters for the equations. Another branch for the study is to consider similar food chain for example if the sharks can eat also a plankton.

References

- [1] Dewdney, A. (1984) “Computer Recreations: Sharks and fish wage an ecological war on the toroidal planet Wa-Tor”, *Scientific American* December pages 14 – 22
- [2] Habib, S.H. (2014) “Computational Ecology: predator-prey model” *Saudi Research Science Institute*
- [3] Williams, W. (1987) “Load balancing and hypercubes: a preliminary look” *Hypercube Multiprocessors*
- [4] Durrett, R. & Levin, S. (2000) “Lessons on Pattern Formation from Planet WATOR” *J. theor. Biol.* 205, 201-214
- [5] Filliben, J. (2013) “NIST/SEMATECH e-Handbook of Statistical Methods”, Available at <http://www.itl.nist.gov/div898/handbook/index.htm>, Last retrieved on 10 November 2015
- [6] Arnold, V. (2000) “Ordinary Differential Equations” *Springer Textbooks*