



AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

KATEDRA INFORMATYKI STOSOWANEJ

Sprawozdanie z modelowania systemu ofiara-drapieżnik

Autor:

Mateusz Wydmański, Michał Kałduś, Rafał Kwaśnik

Kierunek studiów:

Informatyka

Opiekun pracy:

dr inż. Jakub Porzycki

Kraków, 2016

Spis treści

1. Wprowadzenie	3
2. Analiza modelu z zachowaniem stadnym	7
2.1. Model matematyczny	7
2.2. Stabilność	7
2.3. Bifurkacje	8
2.4. Symulacje	9
3. Automaty komórkowe.....	11
3.1. Zagadnienie automatów.....	11
3.2. Implementacja automatu	12
4. Równania Lotki-Volterry dla konkurencji	13
4.1. Model matematyczny	13
4.2. Właściwości modelu.....	14
4.3. Symulacje dla dodatnich współczynników	15
5. Proponowany model zjawiska.....	17
5.1. Cele modelu.....	17
5.2. Działanie algorytmu	17
5.3. Symulacja zjawiska	19
5.3.1. Narzędzi	19
5.3.2. Aplikacja	19
5.3.3. Wyniki	21
5.4. Wnioski.....	26
5.4.1. Nasz model w kontekście istniejących rozwiązań	26
5.4.2. Zebranie najważniejszych wyzwań i trudności rozpatrywanego problemu	26

1. Wprowadzenie

$$\frac{dx}{dt} = \alpha x - \beta xy$$

$$\frac{dy}{dt} = \delta xy - \gamma y$$

gdzie

x – liczebność populacji ofiary

y – liczebność populacji drapieżnika

$\frac{dx}{dt}, \frac{dy}{dt}$ – przyrost populacji w jednostce czasu

t – czas

$\alpha, \beta, \delta, \gamma$ – parametry opisujące interakcje między populacjami i właściwości populacji

Modele ofiar drapieżników są jednymi z najważniejszych elementów składających się na bio- i ekosystemy. Gatunki konkurują, ewoluują, rozprzestrzeniają się dla bołaczki poszukiwania pożywienia i możliwości przedłużenia linii garunku. Konkurencja może przyjmować różne formy: drapieżnik-ofiara, wirus-system immunologiczny, pasożyt-nosiciel, surowiec-konsument, itd.

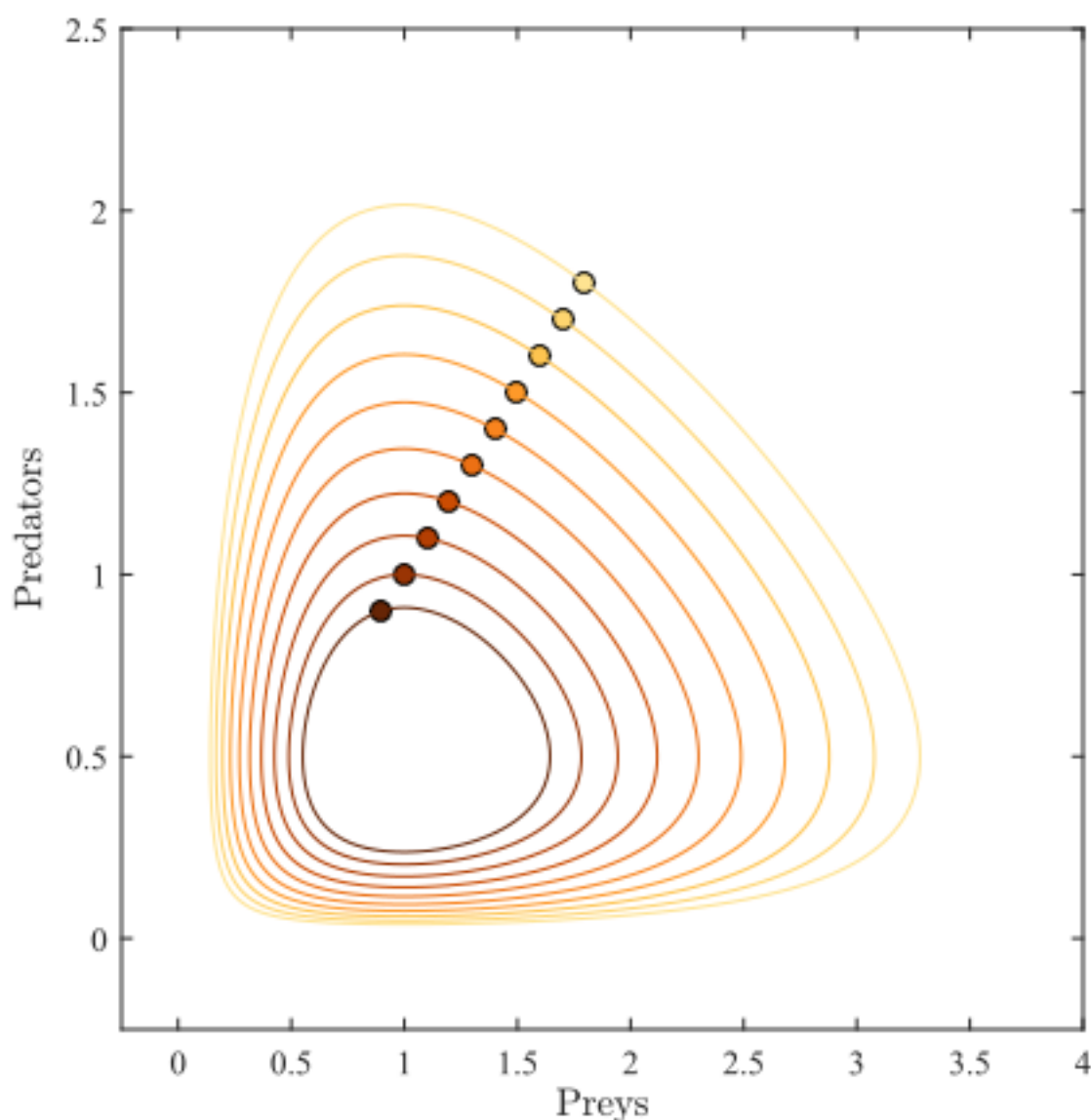
W 1926 słynny Włoski matematyk Vito Volterra zaproponował prosty model opisujący interakcje między ofiarą i drapieżnikiem. Bazuje na kilku założeniach dotyczących środowiska, w którym koegzystują populacje ofiar i drapieżników, to jest:

- Ofiara ma zawsze wystarczająco dużo pożywienia.
- Zaopatrzenie w pożywienie drapieżników zależy wyłącznie od wielkości populacji ofiar.
- Tempo wzrostu populacji jest proporcjonalne do jej wielkości.
- Środowisko nie podlega zmianom na korzyść którejkolwiek z populacji, a ewolucja genetyczna jest niekolekwentna.
- Drapieżniki mają ograniczony apetyt.

Równania Lotki-Volterra, to para równań różniczkowych I rzędu, nieliniowych.

Analiza równań mówi nam, że zmiana ilościowa populacji ofiar jest równa jej tempu wzrostu minus tempo z jaką populacja drapieżników eliminuje ofiary ze środowiska. Z drugiego zaś biegunu, tempo wzrostu populacji drapieżników jest równoważne tempu wzrostu zasilanym przez dostępność pożywienia minus tempo ubytku populacji na skutek śmierci naturalnej.

Portret fazowy modelu Lotki-Volterra daje interesujące rezultaty.



Powyższy diagram pokazuje fluktuacje w ilości populacji ofiar i drapieżników. Koła zaznaczone na brzegach zamkniętych powierzchni pokazują warunki początkowe (dla stałych parametrów), gdzie ilości na osiach są rzędu 10^3 . W tym przypadku punktem stałym (populacje obu gatunków są ustalone i nie podlegają zmianom w czasie) jest punkt $(1; 0.5)$.

W modelu wyraźnie widać, że okres dobrego rozwoju drapieżników przypada gdy populacja ofiar jest duża. Jednakże, poprzez nadmierny rozrost i stopniowe malenie pokładów pożywienia, populacja drapieżników maleje. Wówczas, ofiary nie są na przysłowiowym „celowniku” i mają warunki do rozwoju. Koło się zatoczyło i cykl powtarza się.

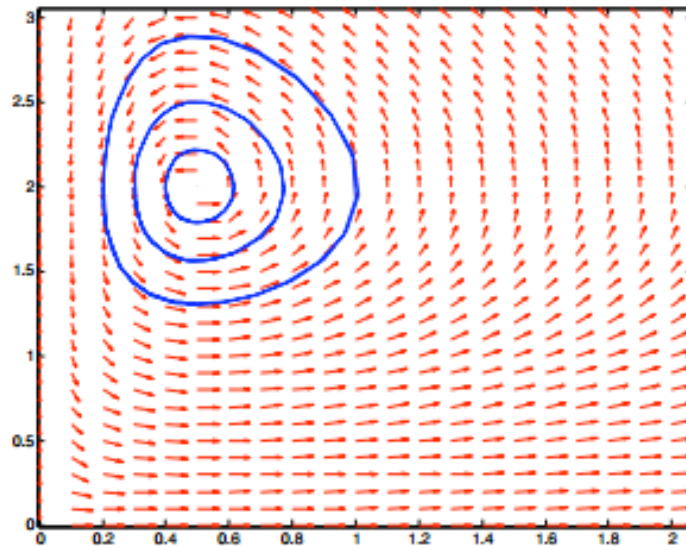
Dla podanego systemu istnieją dwa ekwilibria. Jedno, punkt $(0; 0)$, jest równoznaczne wymarciu obu populacji. Taki stan rzeczy będzie się miał niezależnie od upływającego czasu. Podobnie, drugie ekwilibrium, z obiema populacjami o dodatnich wartościach, jest zależne od początkowych parametrów określających właściwości populacji ofiar i drapieżnika.

Interesujące z biologicznego punktu widzenia jest stabilność układu dla drugiego ekwilibrium.

$$\lambda_1 = i\sqrt{\alpha\gamma} \quad \lambda_2 = -i\sqrt{\alpha\gamma} \quad (1.1)$$

Linearyzacja modelu wokół interesującego punktu: wartości własne jakobianu dla modelu przy drugim ekwilibrium są czysto urojone. Ten krytyczny punkt jest bifurkacją Hopfa – przy niewielkiej zmianie stabilność systemu zmienia się i pojawiają się rozwiązania okresowe. Warto podkreślić, że I metoda Lapunowa nie rozstrzyga o stabilności punktu równowagi, jeżeli system zlinearyzowany jest jedynie stabilny. **Mit**

Punkt ustalony jest eliptyczny i na podstawie teorii Kolmogorowa-Arnolda-Mosera (KAM), która daje odpowiedź na zachowanie w pobliżu punktów eliptycznych, populacje odchylone od punktu ustalonego będą podlegały oscylacjom z częstością $\omega = \sqrt{\alpha\gamma}$.



Rys. 1.1. Portret fazowy przedstawionego modelu. Widać oscylujące wokół ekwilibrium rozwiązania i nieograniczony wzrost jednej populacji przy wymarciu drugiej

2. Analiza modelu z zachowaniem stadnym

2.1. Model matematyczny

Model matematyczny opisujący zachowania ofiara-drapieżnik:

$$\begin{aligned}\frac{\partial X}{\partial t} &= rX\left(1 - \frac{X}{K}\right) - \frac{\alpha\sqrt{XY}}{1 + t_h\alpha\sqrt{X}} + d_x \nabla^2 X \\ \frac{\partial Y}{\partial t} &= sY^2 + \frac{c\alpha\sqrt{XY}}{1 + t_h\alpha\sqrt{X}} + d_y \nabla^2 Y\end{aligned}$$

gdzie

X – liczebność populacji ofiary

Y – liczebność populacji drapieżnika

r – współczynnik przyrostu populacji ofiary

K – zdolność pojemnościowa układu

α – efektywność poszukiwawcza populacji ofiar przez drapieżniki

c – współczynnik żarłoczności drapieżników

t_h – średni czas działania

$\frac{dx}{dt}, \frac{dy}{dt}$ – współczynnik dyfuzji odpowiednich populacji

W powyższym modelu populacje poruszają się losowo - opisane poprzez model ruchów Browna. Zachowanie zostało uwzględnione w równaniach. Czas działania autor opracowania przyrównał do zera dla uproszczenia rozważań. ∇ jest operatorem Laplace'a w przestrzeni dwuwymiarowej $R = (R_1, R_2)$ używanym dla wyrażenia ruchów populacji.

2.2. Stabilność

Interesuje nas stabilność tego systemu. Z biologicznego punktu widzenia jesteśmy zainteresowani w studiach zachowania koegzystujących populacji ofiar i drapieżnika. Niech punkt równowagi będzie postaci (x^*, y^*) , gdzie obie populacje mają dodatnie wartości i żadna nie wymarła. Okazuje się, że takie ekwilibrium istnieje.

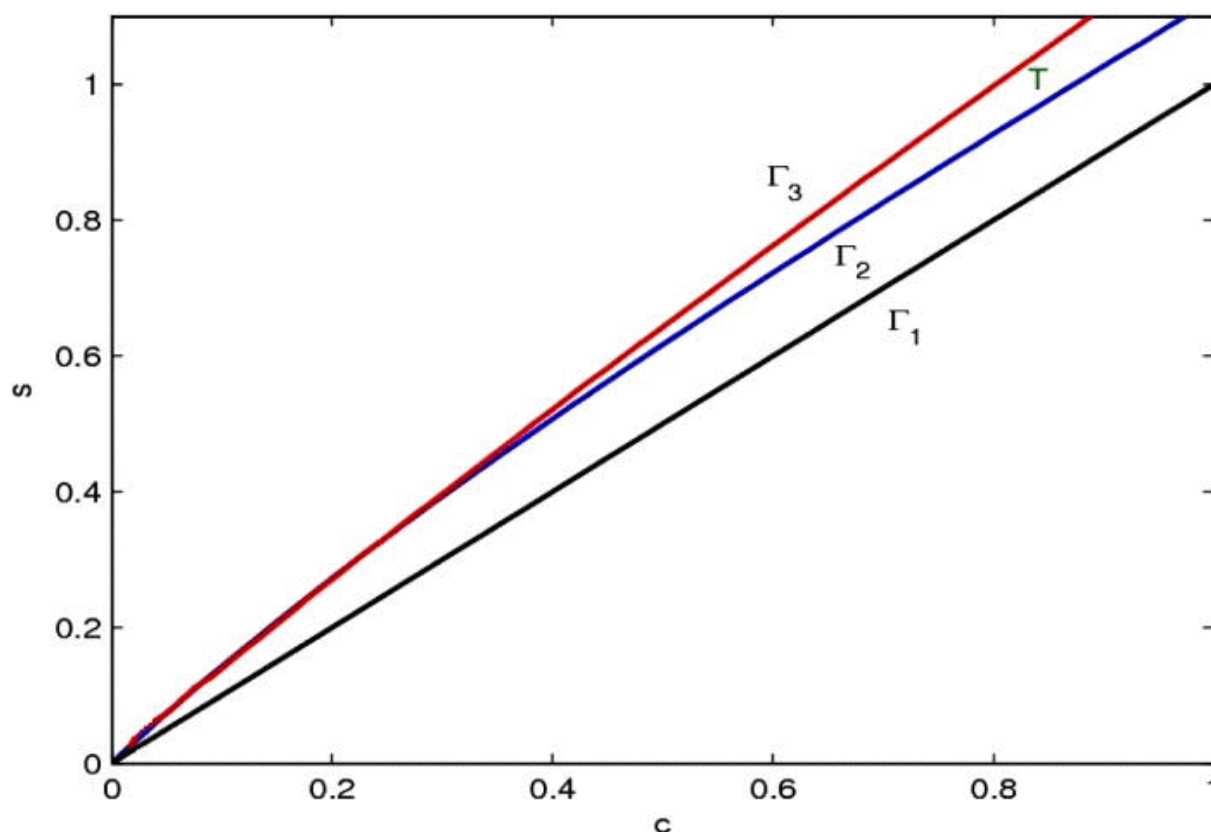
$$x^* = 1 - \frac{c}{s}, \quad y^* = \frac{c}{s} \sqrt{x^*}$$

2.3. Bifurkacje

Wraz ze zmianą niektórym parametrów (c, s) struktura jakościowa modelu może się dramatycznie zmienić. Bifurkacją nazywamy skokową zmianę właściwości modelu matematycznego przy niewielkiej zmianie parametrów. Przykładem może być liczba Reynoldsa, ważna w mechanice płynów bezwymiarowa liczba, która oszacowuje występujący podczas ruchu płynu stosunek sił bezwładności do sił lepkości. Przez zmianę tego parametru ruch płynu może zmienić się z laminarnego w falowy albo turbulentny.

W systemach reakcji-dyfuzji wyróżniamy dwa typy bifurkacji - Hopfa i Turinga. Interesującą bifurkacją jest ta druga, prowadzi bowiem do stanu, gdzie na całej modelowanej przestrzeni pojawiają wzory. W dwóch wymiarach są nimi zwykle heksagony lub figury o pasiastej aparycji.

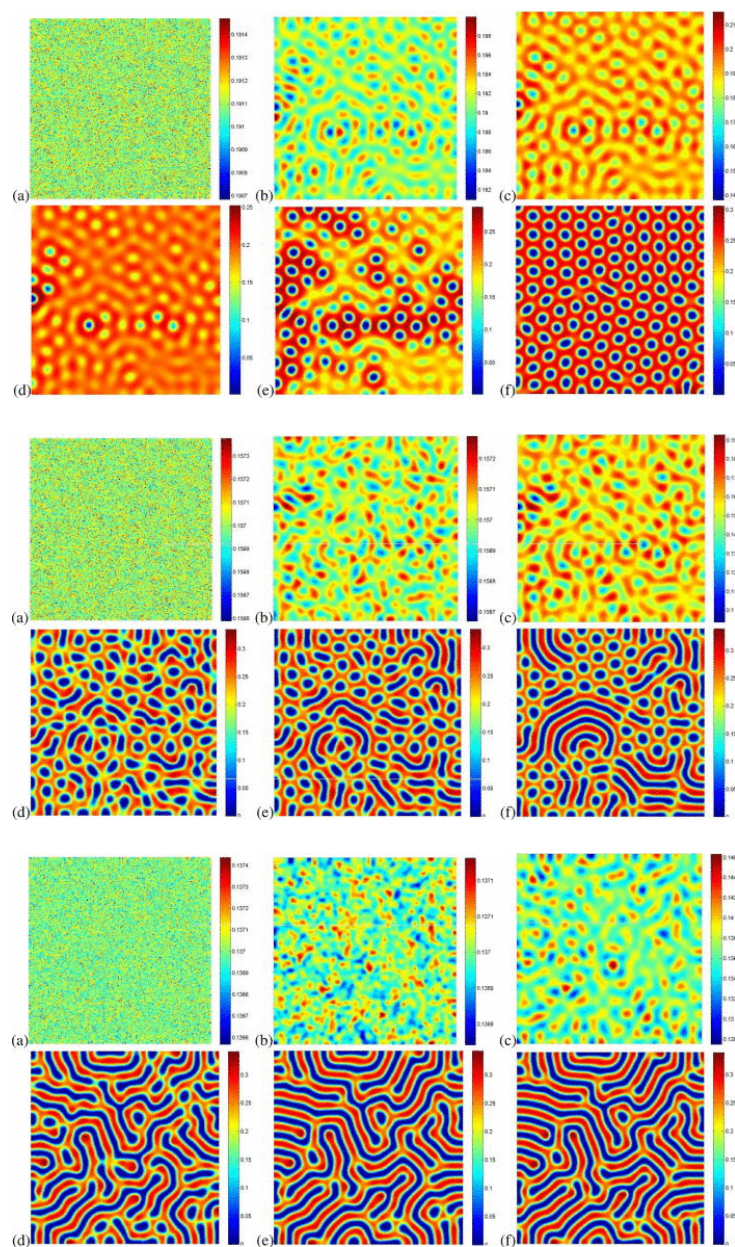
W modelu możliwe jest wyznaczenie relacji między parametrem s i c , dla którego wartość parametru s będzie wartością krytyczną dla bifurkacji Hopfa i Turinga.



Rys. 2.1. Przestrzeń Turinga oznaczona jako T dla $\delta = 10$. τ_1, τ_2, τ_3 są kolejno są liniami bifurkacji Turinga, Hopfa, i ekwilibrium egzystencji.

2.4. Symulacje

Istotnie, interesujące zachowania ujawniają się dla wyznaczonych parametrów. Widoczne są różne kategorie wzorów na skutek bifurkacji Turinga dla różnych wartości parametrów w przestrzeni Turinga.



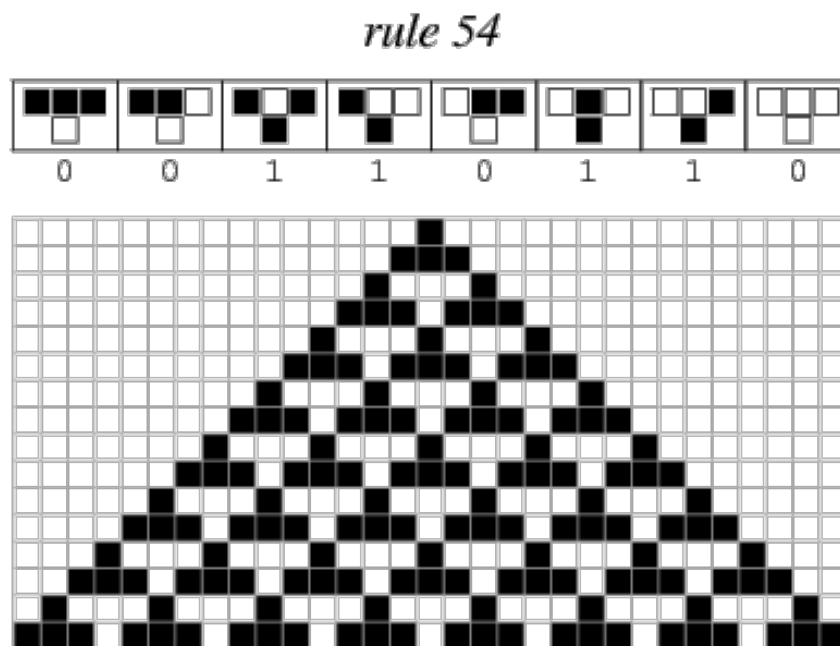
Rys. 2.2. Ewolucja zagęszczenia populacji ofiar w funkcji czasu

3. Automaty komórkowe

3.1. Zagadnienie automatów

Ciekawym zagadnieniem dotyczącym dzisiejszej informatyki jest symulacja różnego rodzaju zjawisk, wydarzeń i populacji. Interesującą właściwością symulowanych modeli jest złożoność zachowań przy nakreśleniu prostych reguł działania. Na myśl nasuwa się pytanie: jakie zachowania będą przejawiały systemy zamodelowane bardziej wyrafinowanymi zasadami. Ograniczenia systemu w połączeniu z modelami matematycznymi oraz mocą obliczeniową komputerów mogą dać ciekawe rezultaty, których moglibyśmy nie przewidzieć. Utworzenie odpowiedniego modelu i wybranie odpowiednich warunków początkowych, pozwala nam też sprawdzić jak zachowa się dany system. W celu zrealizowania podanego systemu jako program komputerowy na pomoc przychodzą nam automaty komórkowe. Dzięki prostocie zasad ich działania w bardzo krótkim czasie jesteśmy w stanie stworzyć prosty mechanizm, który będzie symulował rozwój populacji zwierząt. Stopniowo wprowadzając zasady działania populacji, od najprostszych jakimi jest poruszanie się i zdobywanie pokarmów, po te bardziej skomplikowane jakimi jest rozmnażanie, uciekanie przed drapieżnikami czy polowanie na inne zwierzęta. Naszą uwagę przykuł fakt jak prosty jednowymiarowy automat komórkowy o prostych zasadach funkcjonowania może, przy niewiele różniących się od siebie regułach, dać zupełnie inny obraz wynikowy. Należy też zauważyć, że najwięcej funkcjonalności automatów komórkowych możemy znaleźć analizując właśnie jednowymiarowe automaty. Tutaj definiujemy komórkę oraz jej sąsiednie komórki, które znajdują się w pewnej odległości od niej. Dzięki temu jesteśmy za pomocą funkcji przejścia wyznaczyć następny stan danej komórki. Zaczniemy od tablicy jednowymiarowej i oznaczymy sobie dwa stany: 1 jako komórkę żywą, 0 jako komórkę martwą, a jako sąsiadów przyjmiemy tylko te komórki, które znajdują się bezpośrednio przy niej. Ustalimy też, że następny stan będzie zależał bezpośrednio od stanu samej komórki oraz stanu sąsiadów. Wybierzemy też przejścia takie jak na tablicy.

111	110	101	100	011	010	001	000
0	0	1	0	0	1	1	0



Rys. 3.1. rezultat wykonania operacji wedle tablicy powyżej

3.2. Implementacja automatu

Modelowanie populacji drapieżnik-ofiara za pomocą automatów komórkowych jest złożoną operacją. Każda komórka to zbiór stanów poszczególnych osobników. Każdy następny stan bazuje na poprzednim oraz na stanach sąsiadów.

Autorzy książki *A predator-prey model based on fully parallel cellular automata* przeprowadzi symulację na podstawie populacji wilków i owiec. Udało im się znaleźć trzy możliwe stany, do których będzie dochodził system. Są nimi:

- współistnienie obu gatunków,
- rozwijanie się populacji ofiar,
- wyginięcie obu gatunków

Te stany zależały od ilości pożywiania dla ofiar, czasu po jakim zwierzęta były w stanie się rozmnażać oraz sposobem opiekowania się na młodymi. Największym problemem z jakim spotkali się twórcy było to, że drapieżnik podczas polowania mógł zabić kilka ofiar. Sposobem na poradzenia sobie z tego typu zjawiskiem było wprowadzenie specjalnego sposobu doboru sąsiedztwa, które nie występowała w innych automatach, a pozwala blokować możliwość zabicia wszystkich ofiar. Dodatkowo dyskutowali też jaki wpływ na rozwoju populacji będą mieć takie czynniki jak zagęszczenie oraz mutacje poszczególnych gatunków. Zastanawiali się też jakie wtedy stany może osiągnąć taki automat.

4. Równania Lotki-Volterry dla konkurencji

4.1. Model matematyczny

Rozważając temat modelowania populacji za pomocą równań logistycznych warto poruszyć temat modeli uwzględniających również konkurencję o pożywienie i ogólnie system z wielogatunkowymi poziomami troficznymi.

W odpowiedzi przychodzi układ równań w formie podobnej do klasycznych, modelujących relacje ofiara-drapieżnik. Oparte są na bazie równań logistycznych postaci $\frac{d}{dx}f(x) = f(x)(1 - f(x))$. W przypadku standardowego układu Lotki-Volterry przypadkiem bazowym był wzrost wykładniczy.

Przypadek zająca i owcy mógłby zostać opisany następującym układem równań:

$$\begin{aligned}\frac{dN_1}{dt} &= N_1 r_1 \left(1 - \frac{(N_1 + \alpha_{12} N_2)}{K_1} \right) \\ \frac{dN_2}{dt} &= N_2 r_2 \left(1 - \frac{(N_2 + \alpha_{21} N_1)}{K_2} \right)\end{aligned}$$

r_1, r_2 – wzrost per capita członków poszczególnych populacji

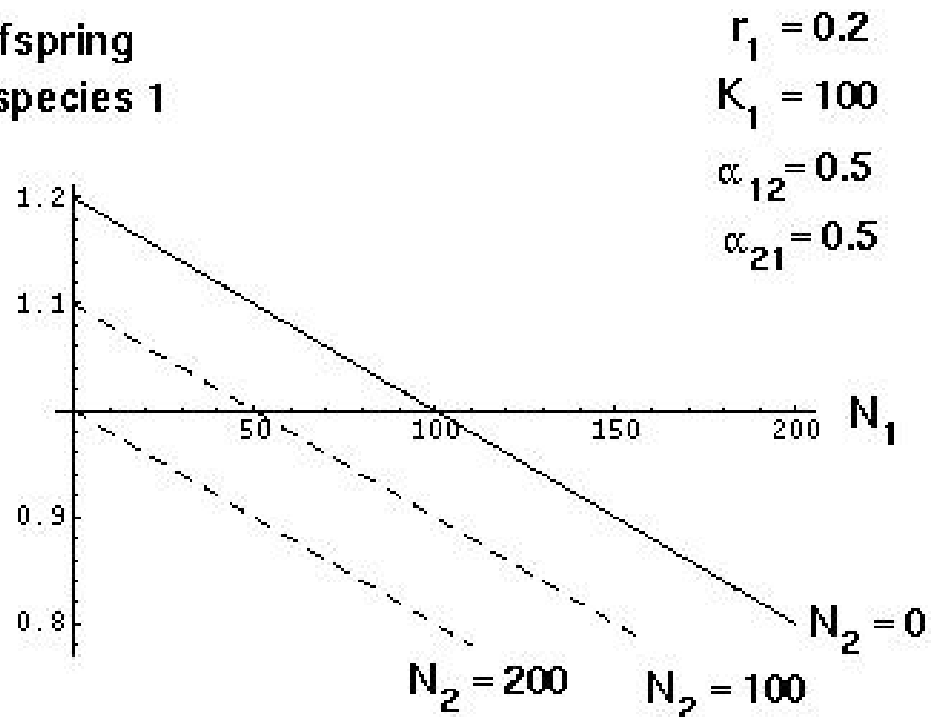
α_{nm} – współczynnik konkurencji jednostki z populacji m na jednostkę z populacji n

K_1, K_2 – zdolność pojemnościowa układu dla poszczególnych populacji

Założeniem logistycznego modelu jest to, że liczba potomstwa per rodzic maleje liniowo ze wzrostem populacji tej populacji.

Włączając w to konkurencję innego gatunku, liczba potomstwa na rodzica zależy nie tylko od populacji pierwszej, ale również od populacji drugiej.

Number of offspring
per parent in species 1

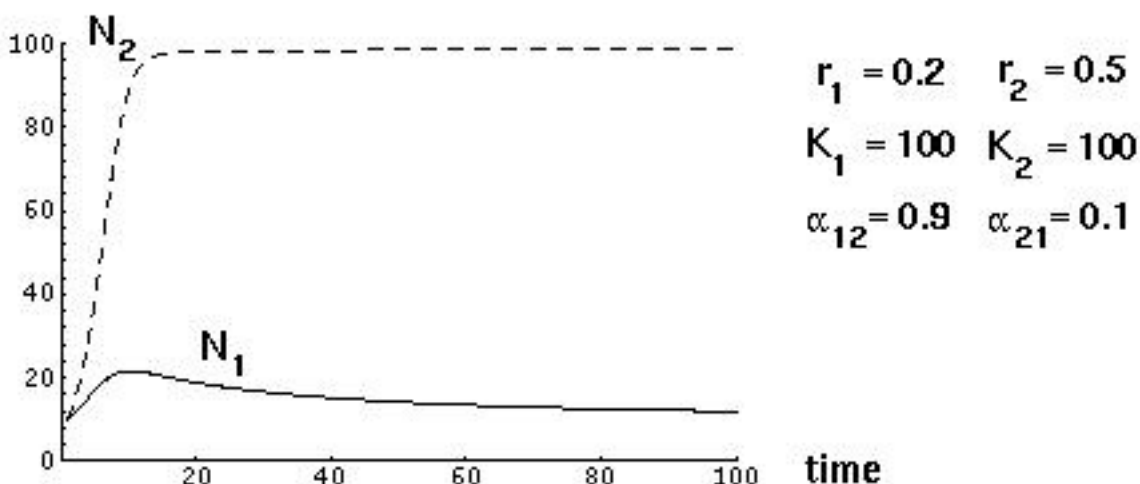
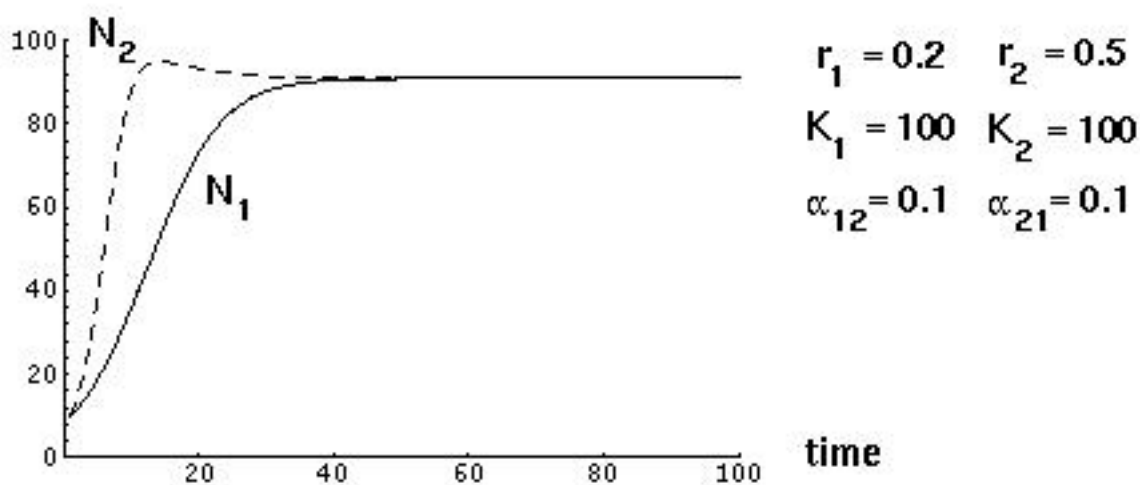


Rys. 4.1. Liczba potomków na dorosłego osobnika w funkcji parametrów

4.2. Właściwości modelu

- jeśli α_{12} wynosi zero, wtedy dynamika gatunku pierwszego będzie przedstawiona równaniem logistycznym (sigmoidalny wykres)
- analogicznie dla $\alpha_{21} = 0$
- jeśli $\alpha_{12} < 0$ to populacja druga zwiększa liczbę surowców dostępnych dla populacji pierwszej
- jeśli oba współczynniki $\alpha_{12}, \alpha_{21} = 0$ to relację między populacjami nazywamy mutualizmem
- jeśli α_{12} albo α_{21} jest równe zero (albo bardzo blisko zera), to mówimy że populacje są w relacji komensalizmu
- jeśli tylko jeden ze współczynników jest ujemny, a drugi dodatni, to mówimy, że populacje są w relacji pasożytnictwa
- jeśli oba współczynniki są dodatnie to populacje ze sobą konkurują

4.3. Symulacje dla dodatnich współczynników



Jeśli α_{12} i α_{21} są małe, to obie populacje osiągają ekwilibrium w pobliżu odpowiedniej pojemności systemu. Jeśli α_{12} jest znacznie większe od α_{21} (gatunek drugi ma większy impakt na liczebność gatunku pierwszego niż odwrotnie), wtedy liczebność gatunku pierwszego jest utrzymywana na niskim poziomie spowodowanym wyższością konkurencyjności drugiego gatunku.

Do stworzenia stabilnego ekosystemu wszystkie wartości własne macierzy a_{ij} muszą być dodatnie. Duże systemy Lotki-Volterry mogą osiągnąć stabilność jeśli współczynniki konkurencji α_{ij} mogą ewoluować zgodnie z naturalną selekcją (**Kon** i **Ack**)

5. Proponowany model zjawiska

5.1. Cele modelu

Celem modelu jest zasymulowanie rozwoju populacji zwierząt. Stworzony przez nas program pozwala na zasymulowanie populacji modelu drapieżnik-ofiara w postaci :

- jednego gatunku ofiary i jednego gatunku drapieżnika, który żywi się ofiarą
- jednego gatunku ofiary, drapieżnika żywiącego się ofiarami oraz drapieżnik drugiego stopnia żywiącego się zarówno ofiarami jak i drapieżnikami, lecz preferującemu drapieżniki.

Zwierzęta mogą poruszać się po zadanym obszarze nie opuszczając go.

Gdy osoba rozpocznie symulację może na bieżąco wyświetlać informacje na temat liczby iteracji, ilości populacji każdego z gatunków. Dodatkowo przedstawiony jest widok mapy migracji zwierząt.

System modeluje zachowanie

5.2. Działanie algorytmu

W pierwszym etapie działania programu generowane są populacje zadanych gatunków. Stworzone populacje są rozmieszczane losowo na mapie w ilości uzależnionej od zadanych przez nas współczynników. Następnie na każdym z pól tworzona jest ilość pożywienia dla ofiar, czyli rośliny.

Po inicjalizacji modelu rozpoczyna się główna część symulacji. W każdej iteracji każde zwierzę obejmuje jakąś strategię działania.

1. Wspólne na wszystkich zwierząt

- Kopulacja
- Wydawanie na świat potomstwa

2. Dla ofiar

- Poszukiwanie i spożywanie roślin
- Ucieczka przed drapieżnikami
- Odpoczynek

3. Dla drapieżników

- Polowanie na ofiary
- Ucieczka przed drapieżnikami drugiego stopnia

4. Dla drapieżników drugiego stopnia

- Polowanie na ofiary i drapieżników

Poszczególne zachowania uzależnione są od aktualnego stanu danego zwierzęcia oraz jego otoczenia.

1. Wspólne strategie zwierząt

- Jeżeli samica jest w kluczowym stadium ciąży musi wydać na świat potomstwo.
- Jeżeli zwierzę jest zagrożone stara się uciec.
- Jeżeli współczynnik głodu jest krytyczny ofiara usiłuje zdobyć pokarm.
- Jeżeli współczynnik pożądania seksualnego jest krytyczny i osobnik jest wystarczająco dojrzały rozmnaża się z drugim przedstawicielem swojego gatunku przeciwnej płci na danym polu lub szuka partnera w swojej okolicy.
- Jeżeli zwierze ma wszystkie współczynniki w normie, odpoczywa.

2. Szczególne dla ofiary

- Jeżeli na polu jest drapieżnik, który poluje na ofiarę stara się uciec z danego pola
- Jeżeli współczynnik głodu jest krytyczny ofiara żywi się pokarmem na aktualnym polu lub jeśli pole jest pozbawione roślin ofiara szuka ich w swoim najbliższym otoczeniu.

3. Dla drapieżnika

- Jeżeli na jego polu jest drapieżnik drugiego stopnia, drapieżnik stara się przemieścić w dogodniejsze dla siebie miejsce
- Jeżeli współczynnik głodu jest krytyczny drapieżnik poluje na ofiarę znajdującą się na jego polu lub szuka jej w swoim najbliższym otoczeniu.

4. Dla drapieżnika drugiego poziomu

- Jeżeli współczynnik głodu jest krytyczny drapieżnik drugiego stopnia w pierwszej kolejności stara się upolować drapieżnika, który jest preferowanym gatunkiem. W razie jego braku poluje na przedstawiciela gatunku reprezentowanego przez ofiarę.

Dodatkowo w celu organiczna ekspansji populacji ofiar liczba roślin na mapie jest ograniczona przez współczynniki regeneracji roślin po każdej iteracji. Po każdej iteracji zwiększany jest wskaźnik głodu dla każdego zwierzęcia oraz pożądania seksualnego dla dojrzałych osobników. Zwierzę po spożyciu pokarmów otrzymuje wartość energetyczną zjadanego pokarmu, która obniża wskaźnik głodu.

5.3. Symulacja zjawiska

5.3.1. Narzędzi

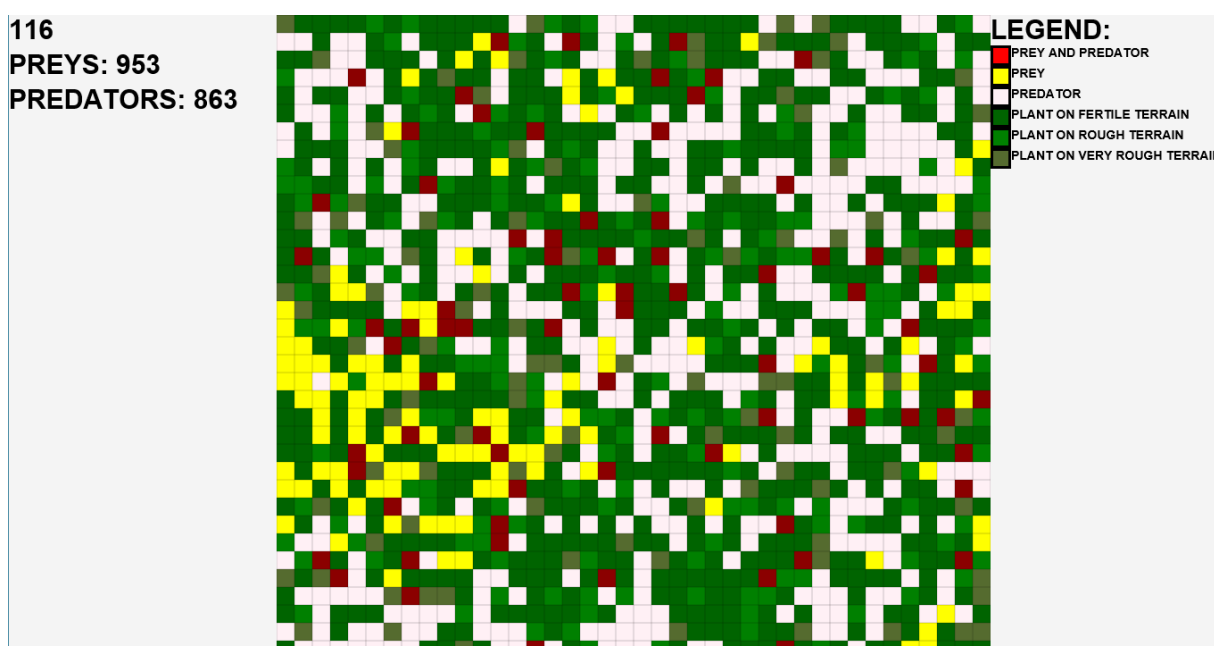
Na potrzeby symulacji została stworzona aplikacja okienkowa, która wizualizuje efekty działania programu. Do zaimplementowania programu posłużyliśmy się językiem Java, a do wyświetlenia danych użyliśmy biblioteki graficznej JavaFX.

5.3.2. Aplikacja

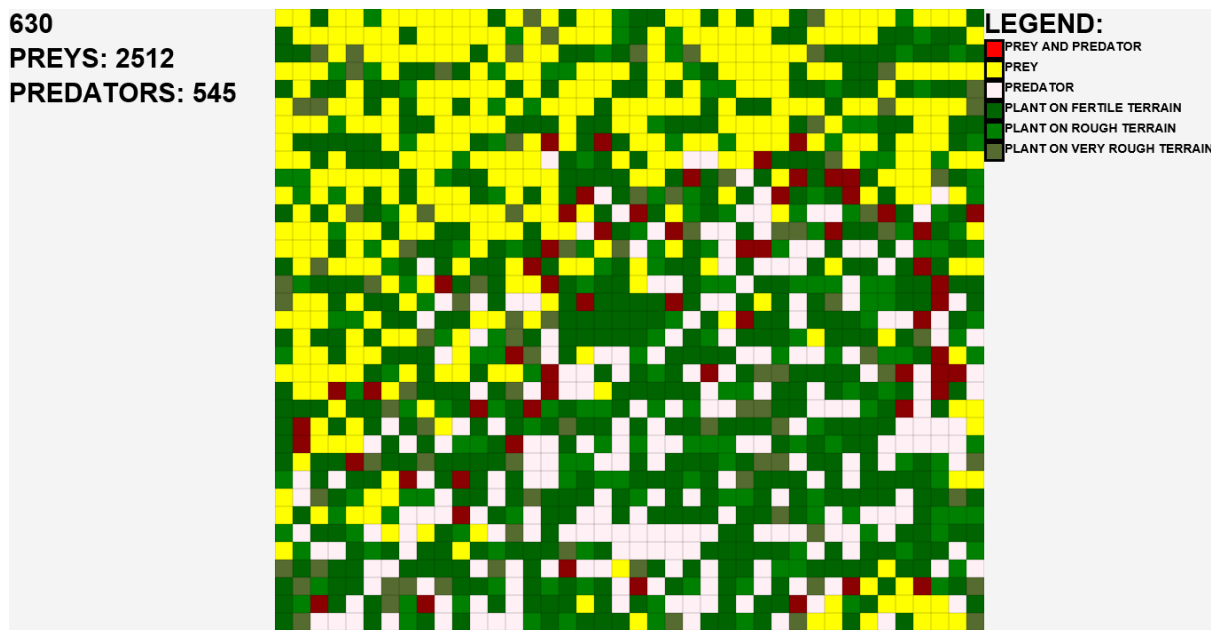
Poniżej przedstawiamy kilka zrzutów ekranu w trakcie działa aplikacji.

5.3.2.1. Symulacja dla 2 gatunków

Poniższe obrazy prezentują zachowanie się aplikacji dla dwóch gatunków zwierząt. Na rysunku 5.1 widać jak ofiary i drapieżniki żyją w pomieszanych stadach (początkowe iteracji), a na rysunku 5.2 widzimy jak uformowały się stada poszczególnych populacji.



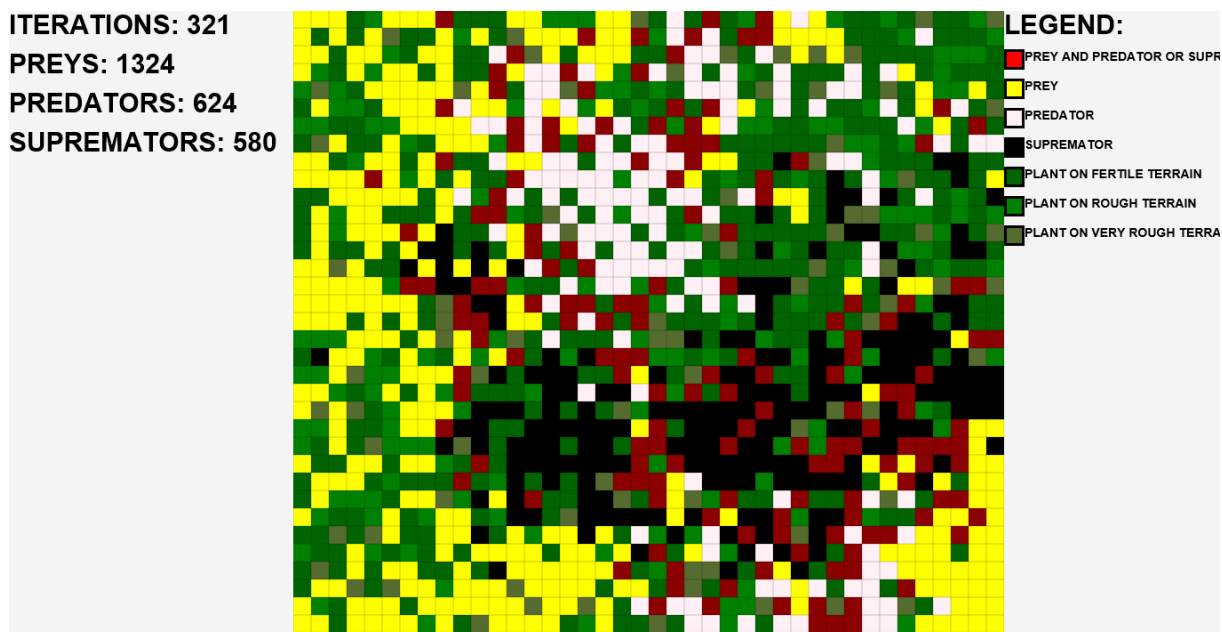
Rys. 5.1



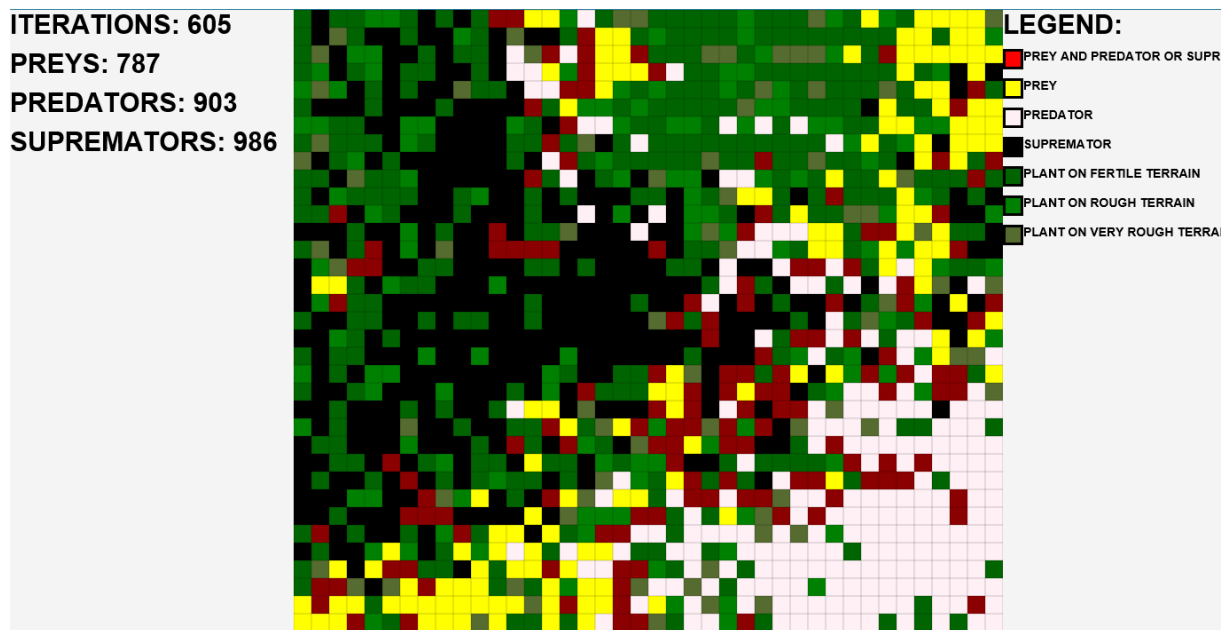
Rys. 5.2

5.3.2.2. Symulacja dla 3 gatunków

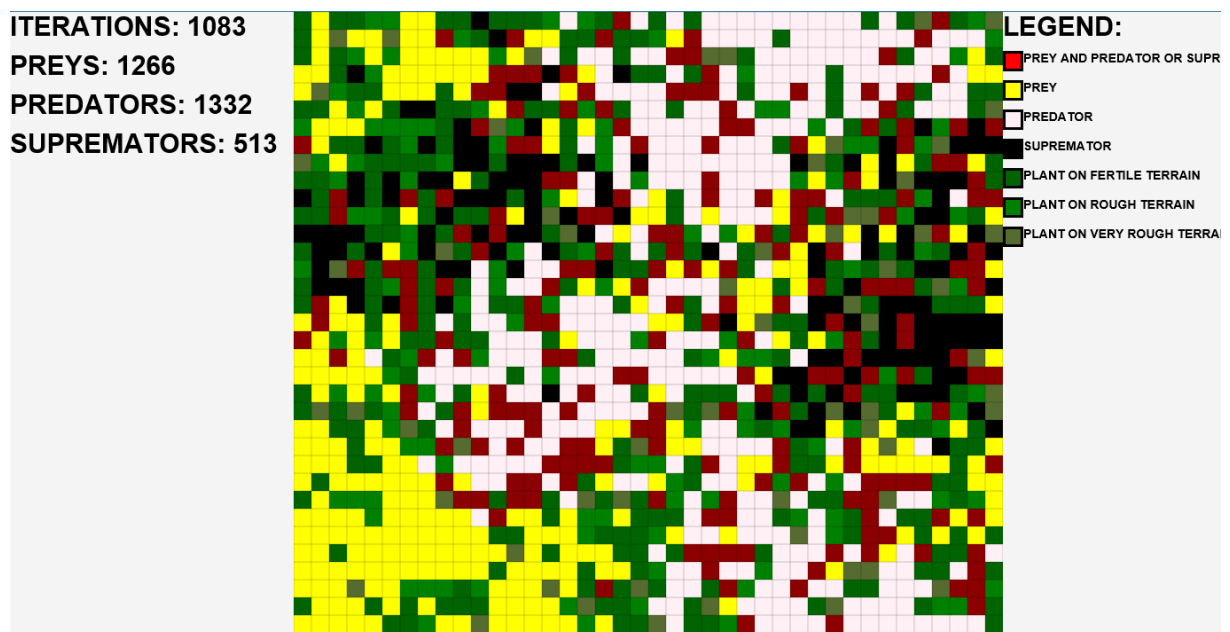
Poniższe obrazy prezentują zachowanie się aplikacji dla trzech gatunków zwierząt (ofiara, drapieżnik i drapieżnik drugiego stopnia). Na rysunku 5.3 widać jak ofiary i drapieżniki żyją w pomieszanych stadach (początkowe iteracji), a na rysunkach 5.4 5.5 widzimy jak uformowały się stada poszczególnych populacji.



Rys. 5.3



Rys. 5.4



Rys. 5.5

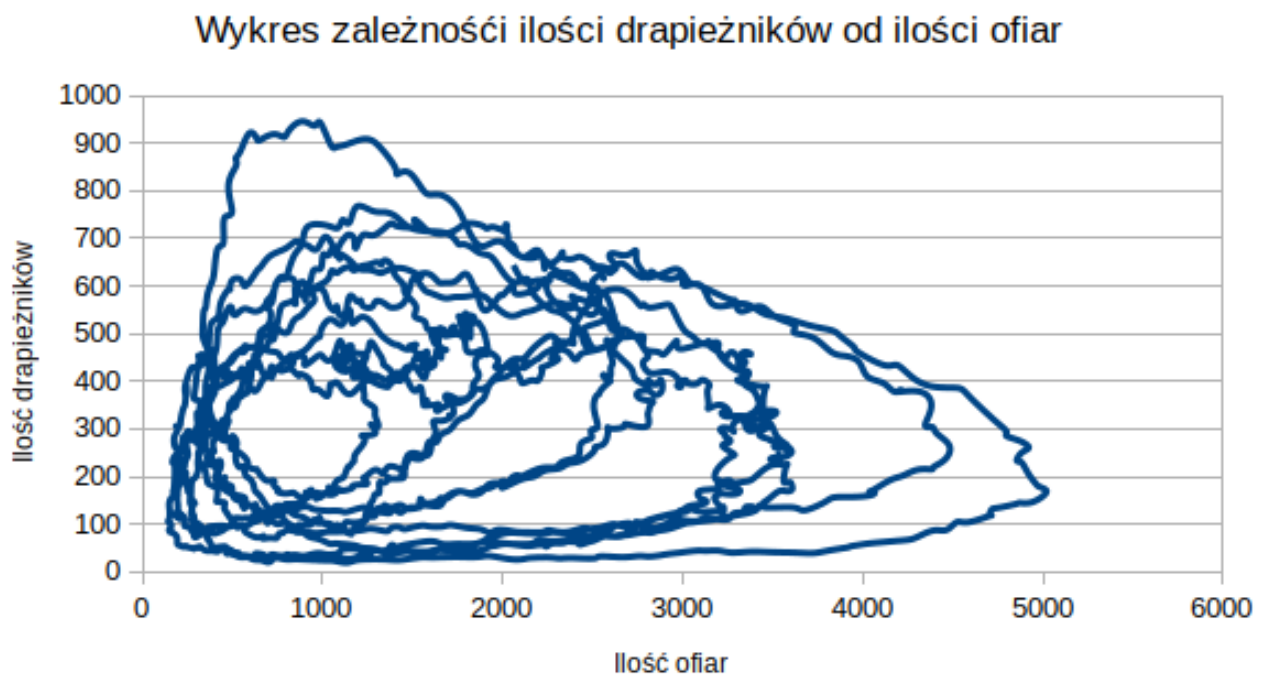
Warto też zwrócić uwagę na to że życie w większych stadach z dala od swoich ofiar powoduje zagrożenie gatunku i przymus migracji.

5.3.3. Wyniki

Wyniki kilku symulacji zapisaliśmy do pliku tekstowego, a następnie stworzyliśmy na tej podstawie odpowiednie wykresy. W celu późniejszej możliwości walidacji przyjętego przez nas modelu. Rysunek

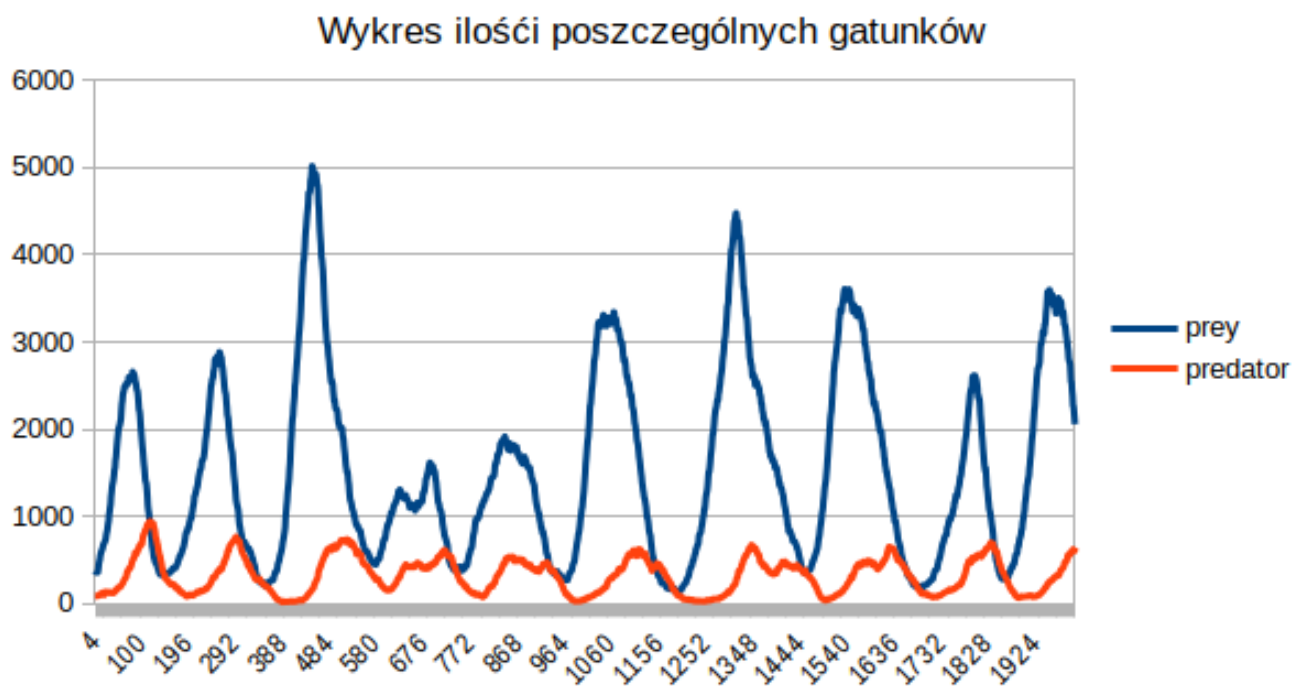
1.1 przedstawia wykres zależności ilości drapieżników od ilości ofiar w trakcie próby liczącej 2000 iteracji. Dodatkowo należy dodać że wszystkie symulacje uruchamiane były z takimi samymi parametrami, a mimo to otrzymaliśmy inne rezultaty.

5.3.3.1. Wyniki dla 2 gatunków- ofiara, drapieżnik

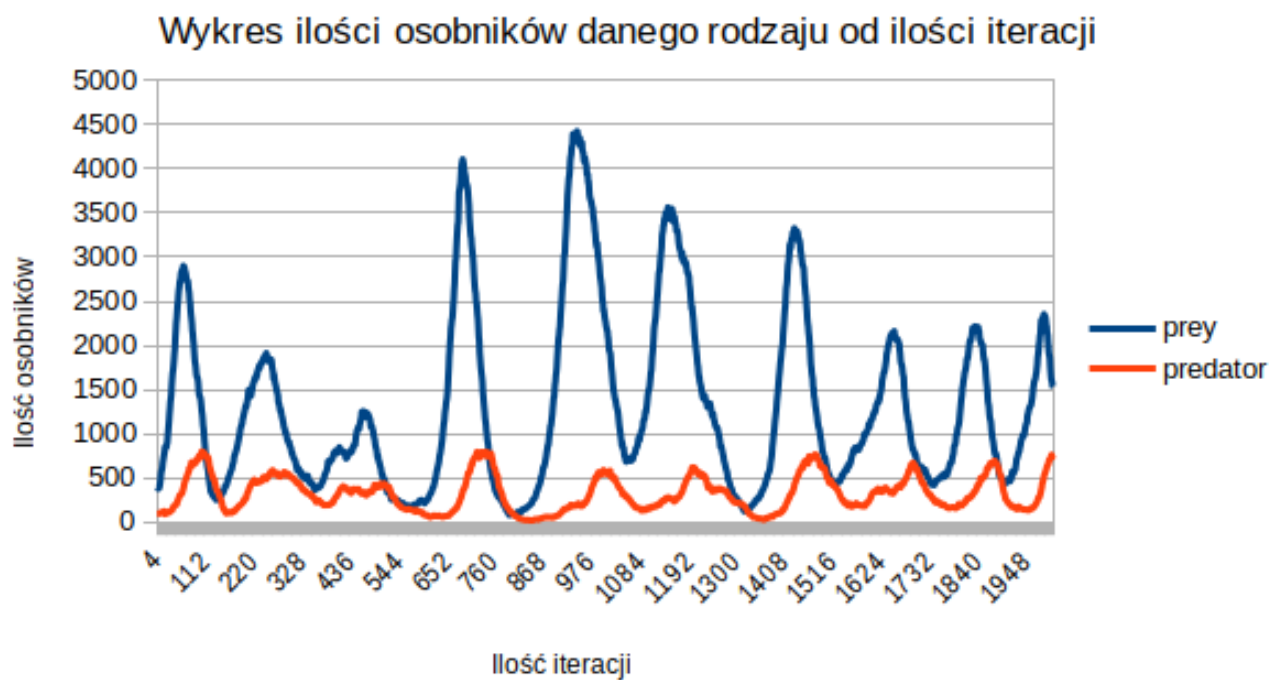


Rys. 5.6

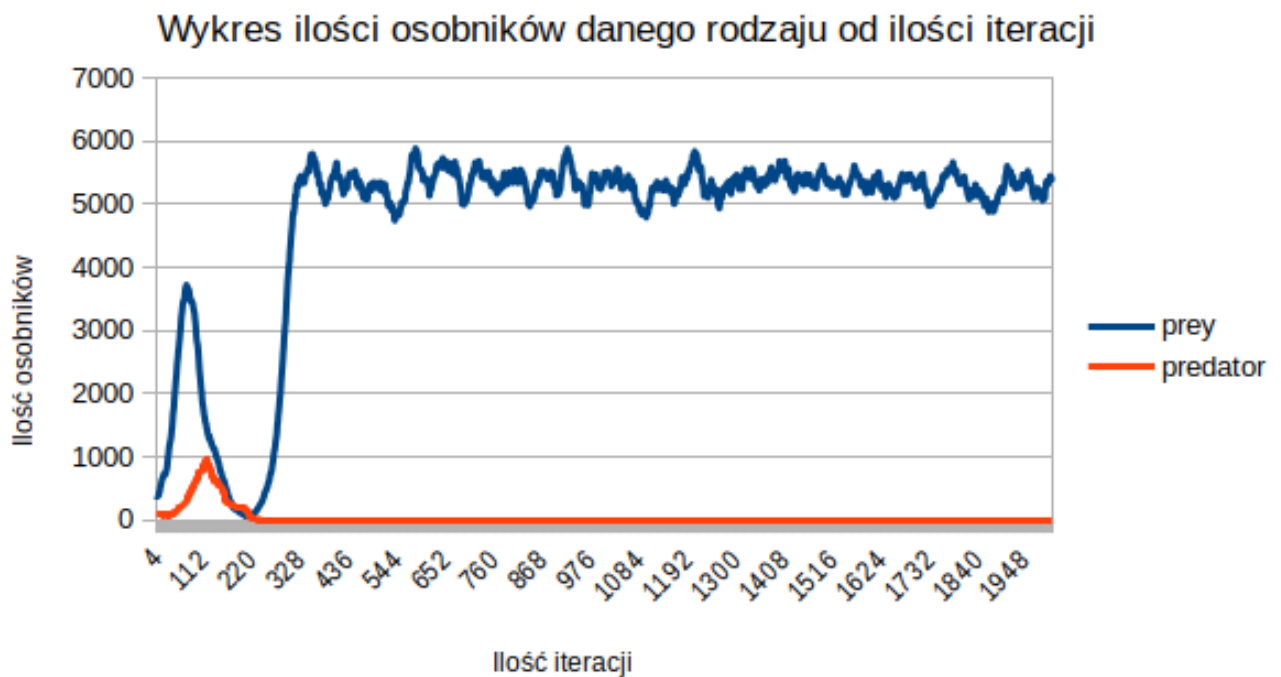
Powyższy wykres zależności ilości drapieżników od ofiar jest zbliżony do portretu fazowego Lotki-Voltery.



Rys. 5.7

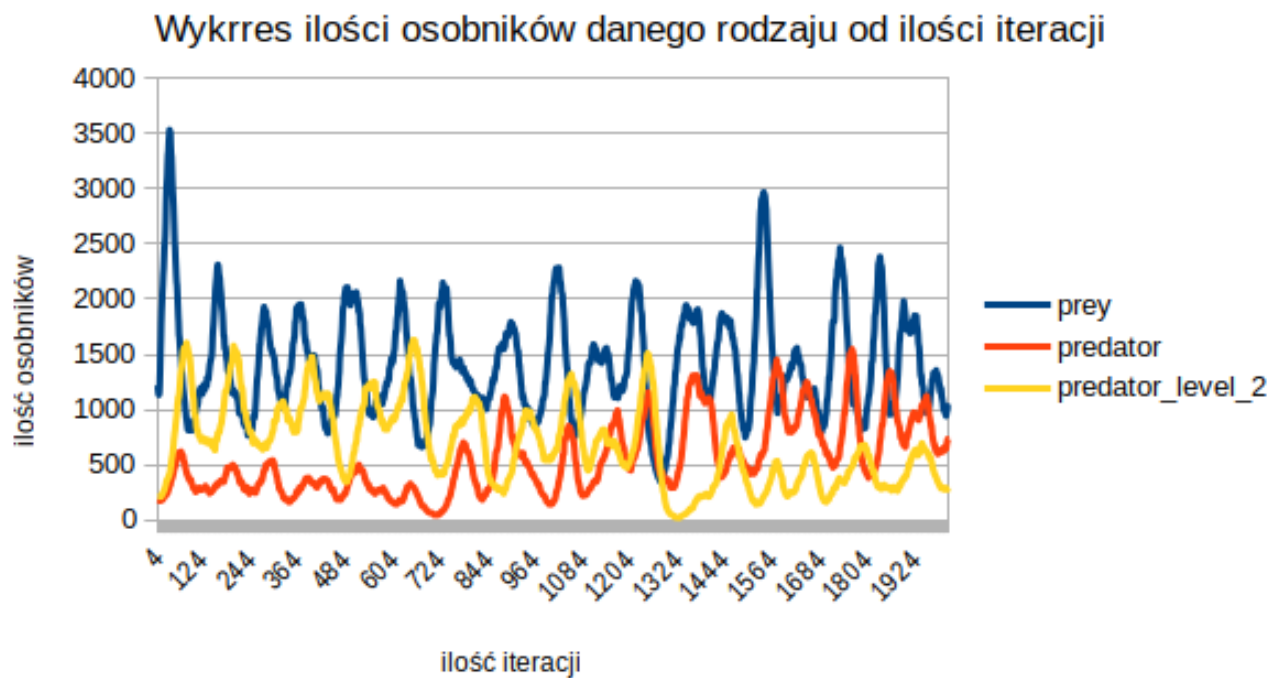


Rys. 5.8

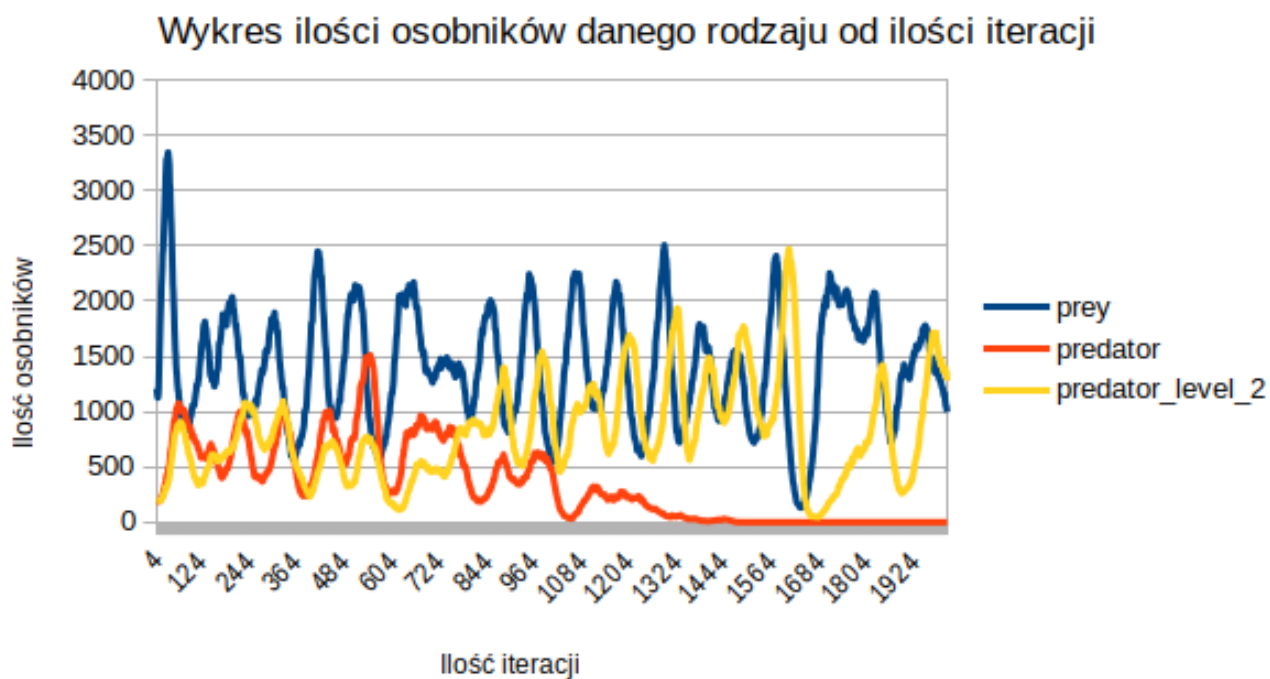


Rys. 5.9

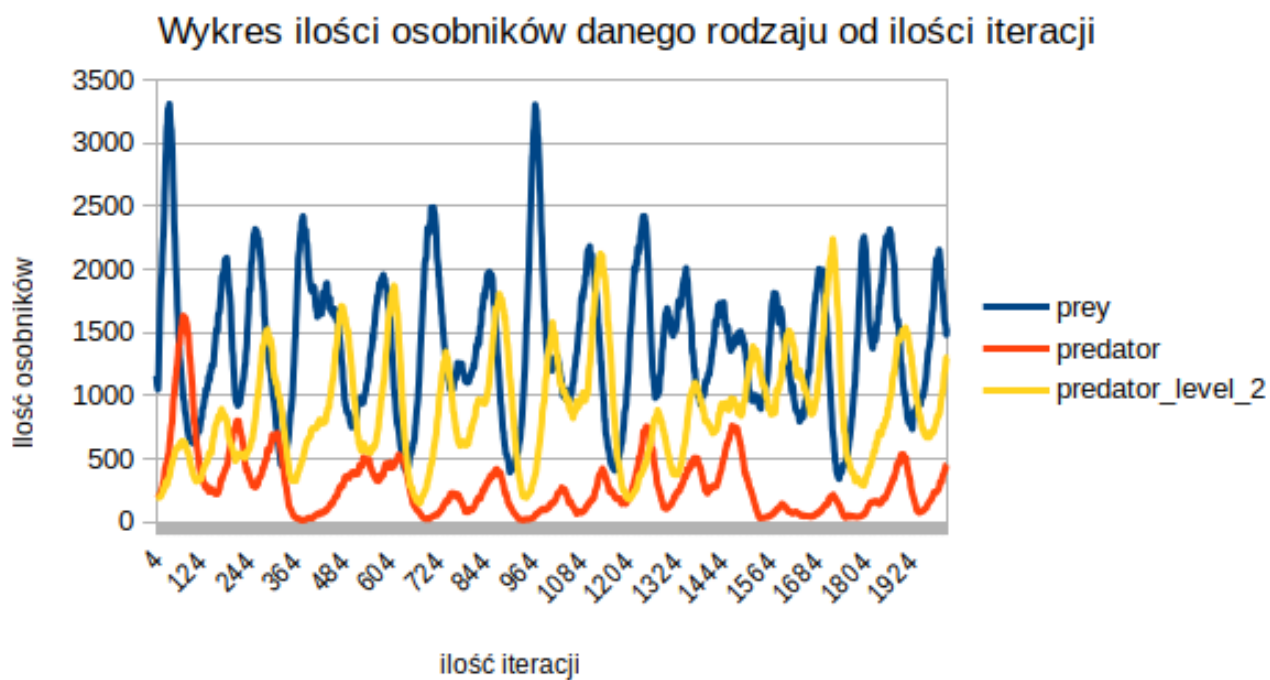
5.3.3.2. Wyniki dla 3 gatunków- ofiara, drapieżnik i drapieżnik drugiego stopnia



Rys. 5.10



Rys. 5.11



Rys. 5.12

5.4. Wnioski

Podstawowym wnioskiem jaki zauważyliśmy podczas tworzenia projektu jest to że część współczynników ma ogromny wpływ na działanie i rezultat symulacji, a część znikomy. Głównymi współczynnikami naszej symulacji są:

- szansa upolowania ofiary
- maksymalna możliwa do pokonania odległość w trakcie jednej iteracji
- przyrost współczynnika głodu

Wpływ na wyniki ma też miejsce ulokowania poszczególnych gatunków. Jeżeli już na samym początku drapieżniki otaczają populację ofiar to oba gatunki wyginą po upływie kilkunastu iteracji.

W sytuacji kiedy nasza populacja ofiar w początkowej fazie działania aplikacji da się zagonić w krawędzie, dostępnego pola, podczas pościgu przez gatunki drapieżników i odłączy się od głównej populacji tylko niewiele stado ofiar istnienie bardzo duże prawdopodobieństwo, że populację ofiar wyginą, a co skutkuje śmiercią populacji drapieżników.

W sytuacji gdy wyginą wszystkie drapieżniki populacja ofiar nie rośnie w nieskończoność, ponieważ istnieje skończona ilość pożywienia dla nich. Wielkość populacji samych ofiar rośnie do pewnej ilości a następnie oscyluje wokół niej.

5.4.1. Nasz model w kontekście istniejących rozwiązań

Po przejrzeniu dostępnej literatury zarówno polskiej jak również angielskojęzycznej można zauważyć, że jest możliwe znalezienie literatury związanej z automatami komórkowymi oraz z rozwojem populacji zwierząt co sprawiło, że podczas tworzenia naszego projektu, łatwiej nam było stworzyć odnowieni model. Mimo to częściowo opieraliśmy się na naszej intuicji przy korzystaniu z różnego rodzaju przykładowych prac związanych z automatami, bardzo często odrobinę odbiegającymi od naszego tematu, ale związanymi z automatami co w dużej mierze pomogło nam podczas wykonywania projektu. Po przetestowaniu naszego projektu korzystając z wielu różnych parametrów wydaje nam się, że całkiem dobrze on przedstawia symulację rozwoju populacji zwierząt w modelu prey-predator i w pełni spełnia on wszelkie ustalone na początku założenia.

5.4.2. Zebranie najważniejszych wyzwań i trudności rozpatrywanego problemu

Pierwszym napotkanym przez nas problemem było ustawienie maksymalnej możliwej do pokonania odległości w trakcie jednej iteracji tak aby jak najbardziej odzwierciedlało to rzeczywistość, bo długich próbach wybraliśmy odpowiednią wartość, co zostało opisane kilka stron wyżej i również przyniosło bardzo dobre rezultaty.

Zatrzymał nas również dylemat związany z szybką ekspiacją populacji drapieżników, która dosyć szybko zabijała populacje ofiar, a następnie sama ginęła. W celu zniwelowania tej niedoskonałości zmniejszyliśmy szanse upolowania ofiary ze 100 % na wartość niższą.

Ważnym elementem jest również to, że w zależności od tego czy dany osobnik jest głodny lub spragniony rozmnażania się wykonuje różne czynności, co skutkuje tym, że polucja migruje oraz zbiera się w grupy w celu tworzenia potomstwa.

Bardzo zależało nam na wykonaniu symulacji, która przedstawi rozwój populacji 3 gatunków co ostatecznie udało nam się uzyskać i jest przedstawione na screenach znajdujących się kilka stron wyżej. To właśnie symulacja rozwoju trzech populacji sprawdziła czy sposób rezliacji programu jest dobry i czy może być w przyszłości rozszerzalne oraz jest bardzo cennym osiągnięciem naszego projektu, przedstawiającym jego prawidłowe działanie.

Ostatecznie udało nam się stworzyć aplikację, która symuluje rozwój populacji zwierząt przy wykorzystaniu wiedzy na temat obszaru, na której się ona znajduje i w zależności od panujących warunków następuje rozwój populacji w dobrych warunkach (wszystkie gatunkom udaje się przeżyć, żaden nie dominuje) jej wymieranie w niekorzystnych warunkach (związanych z wyginieniem ofiar), migracje i ewentualne przemieszczenie się [populacji] w rejony bardziej odpowiednie do rozmnażania się oraz dostępu pokarmu. Warto tu podkreślić, że nie zawsze każda populacja jest w stanie przeżyć, jeżeli znajdzie się ona w obszarze niekorzystnych warunków.