

Rozdział 4

Rozwiązania modelu opisującego rozwój podpopulacji nietoperzy – analiza przy użyciu interaktywnego arkusza programu Maple

Anna Poskrobko⁷

Wydział Informatyki, Politechnika Białostocka

Streszczenie: W artykule przedstawiono analizę zachowania rozwiązań matematycznego modelu opisującego rozwój populacji nietoperzy. Opisano szczegółowo założenia modelu oraz przyrodnicze motywacje jego konstrukcji. Między innymi uwzględniono specyficzną strategię wyszukiwania kryjówek, ograniczony zasięg detekcji nietoperzy w trudnym, pełnym przeszkód, środowisku oraz umiejętność przekazywania informacji w grupie. Zachowanie populacji nietoperzy oraz ich strategia wyszukiwania kryjówek została opisana w pracy po obserwacji zachowań dwóch gatunków nietoperzy zamieszkujących Puszczę Białowieską. Prezentowany model oparty jest na układzie równań różniczkowych zwyczajnych. W pracy przedstawiamy wybrane symulacje rozwiązań w oparciu o interaktywny arkusz programu Maple. Taki interaktywny arkusz pozwala analizować zachowanie rozwiązań numerycznych przy dynamicznej zmianie wielu parametrów modelu. W pracy opisano szczegółowo konstrukcję oraz działanie takiego arkusza, w którym użyto programowalnych, wbudowanych komponentów programu Maple.

Słowa kluczowe: arkusz interaktywny Maple, nietoperze, model matematyczny

Wprowadzenie

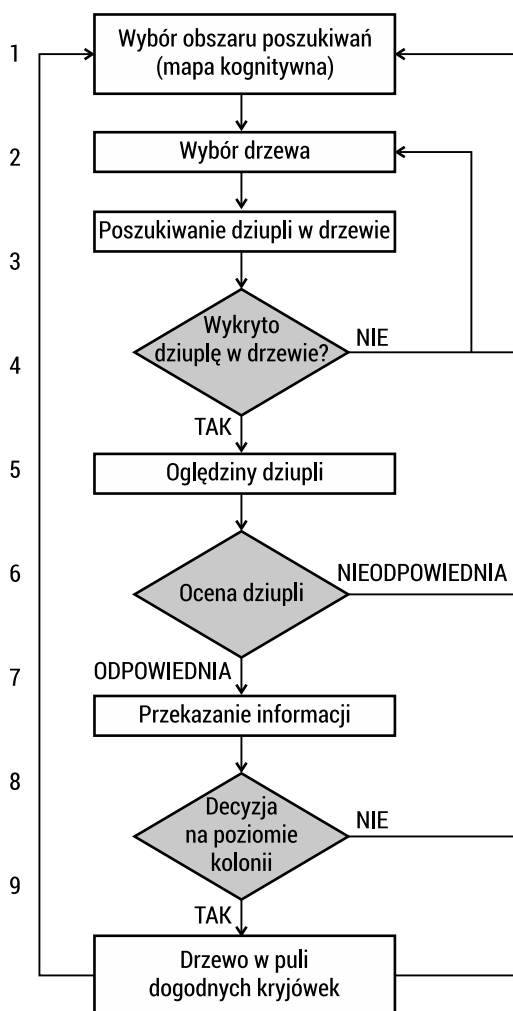
Prezentowany w pracy model odzwierciedla zachowanie oraz strategię wyszukiwania kryjówek dwóch gatunków nietoperzy zamieszkujących Puszczę Białowieską: borowca wielkiego (*Nyctalus noctula*) oraz borowca leśnego lub inaczej borowiaczka (*Nyctalus leisleri*). Informacje zawarte w pracy, dotyczące strategii wyszukiwania kryjówek zaczerpnięte są głównie z artykułu [4].

⁷ Praca została zrealizowana w ramach projektu WZ/WI-IIT/1/2020 finansowanego przed Ministerstwo Nauki i Szkolnictwa Wyższego.

Większość nietoperzy nie potrafi samodzielnie budować schronień. Ssaki te w związku z tym zasiedlają obszary obfitujące w miejsca żerowania oraz różnego rodzaju kryjówki (naturalne lub stworzone przez człowieka). Oba wspomniane gatunki zasiedlają zazwyczaj dziuple drzew. Nietoperze, niezależnie od wieku, przez całe swoje życie poszukują kryjówek. Jedna kolonia nietoperzy może w ciągu roku użytkować nawet kilkanaście kryjówek, zmieniając je co jakiś czas, transportując przy tym swoje młode. Niektóre gatunki nietoperzy tworzą też kolonie rozrodcze, które przenoszą się na nowe miejsca co kilka dni. Pozwala to unikać pasożytów, rozwijających się w dziuplach, a także zmniejszyć ryzyko drapieżnictwa. Duże kolonie (składające się z osobników obu płci) ssaki te tworzą zimą. Przy czym, niektóre gatunki nietoperzy dokonują migracji także o tej porze roku, przerywając stan hibernacji. Dzieje się tak w czasie długotrwałych ociepleń. Nietoperze poszukują wówczas chłodniejszych kryjówek, gdzie niższa temperatura pozwala zredukować wszystkie funkcje życiowe i ograniczyć zużycie tkanki tłuszczowej. Zatem nieustanna migracja jest cechą charakterystyczną życia wszystkich gatunków nietoperzy. Wydaje się zatem słuszne, budując model opisujący zachowanie nietoperzy, skupić się głównie na tym aspekcie ich życia. Pomimo, że nietoperze tuż po gryzoniach są najliczniejszym pod względem liczby gatunków, rzędem ssaków, to jak podkreślają wszyscy badacze, zwyczajnie nietoperzy przede wszystkim ze względu na skryty tryb życia, są jednak jeszcze stosunkowo słabo zbadane. Najlepiej opisane jest zjawisko migracji. Chociaż i tutaj dane dotyczące samej strategii wyszukiwania przez nietoperze pojedynczych dziupli pochodzą z badań laboratoryjnych i eksperymentów ze sztucznymi kryjówkami [3, 4, 5, 6]. Zgodnie z tymi badaniami nietoperze wybierają jedynie miejsca, które dają szansę na dalszy rozwój. Ponadto, ssaki te posiadają umiejętność przenoszenia zdobytych informacji wśród osobników kolonii [1], a także wykazują się pamięcią dotyczącą rozkładu siedlisk w terenie (która nie słabnie pomimo hibernacji i pozwala wrócić do raz wykrytej kryjówki) [8], a maksymalny zasięg wykrywania nowych dziupli to 90 m. Te wszystkie informacje pochodzące z obserwacji rzeczywistych populacji nietoperzy wykorzystano tworząc model oraz wykonując symulacje komputerowe. Zgodnie ze strategią wyszukiwania poszczególnych siedlisk przez pojedyncze osobniki [4], (rysunek 4.1) nietoperz: 1) wybiera teren w pobliżu kryjówki, który penetruje w poszukiwaniu nowego siedliska, 2) znajduje potencjalne drzewo wykorzystując echolokację, wizję oraz uzyskaną wcześniej wiedzę nt. przestrzennego rozmieszczenia drzew, 3) wyszukuje zagłębienia w wybranym drzewie, 4) sprawdza wnętrza zagłębienia, 5)–6) ocenia jego jakość, 7) przenosi informację do kolonii o obecności odpowiedniej dziupli, 8) na podstawie tych informacji kolonia lub jej część decyduje się na przenosiny lub nie, 9) wykryta dziupla wchodzi do zbioru znanych kryjówek, które mogą być użyte ponownie w przyszłości (pamięć).

Nasz model opisuje układ zamknięty (ze skończoną liczbą możliwych siedlisk). W przyrodzie oczywiście taki model nie występuje, w rzeczywistych systemach zawsze znajdują się możliwości dalszej dyspersji populacji. Według opisywanego teoretycznego układu zamkniętego, gdy liczba wszystkich osobników zaczyna zbliżać

się do poziomu sumy pojemności wszystkich kryjówek układ przestaje dobrze opisywać stan rzeczywisty spotykany w przyrodzie. Nadwyżka osobników przestaje być uwzględniona w modelu.



RYSUNEK 4.1. Strategia wyszukiwania kryjówek przez nietoperze i pozyskiwanie informacji o występowaniu i rozmieszczeniu dziupli w lesie

ŹRÓDŁO: [4]

Model matematyczny przedstawiony w pracy opisuje rozwój kolonii nietoperzy i jej podział na podpopulacje. Przy konstrukcji modelu uwzględniono uwagi biologiczne opisane wyżej, czyli szczególną strategię wyszukiwania kryjówek, ograniczony zasięg detekcji oraz dyspersyjny charakter rozwoju kolonii. Model oparty jest na układzie równań różniczkowych zwyczajnych. Większość informacji biologicznych

zaczepnięto z pracy [4], która poza częścią zawierającą opisy biologiczne opisuje także symulację komputerową w języku programowania C++. Symulacja ta obrazuje strategię wyszukiwania dziupli przez pojedynczego osobnika w terenach o różnych topografiach. Natomiast praca [7] prezentuje podejście teoretyczne i używa narzędzi matematycznych (równań różnicowych) do opisu zachowania i orbit ruchu pojedynczych osobników wyszukujących dziuple. Według naszej wiedzy nie ma innych prac, które opisywałyby przy użyciu narzędzi matematycznych zachowanie całej kolonii nietoperzy i jej dyspersyjno-dyfuzyjny charakter.

Model przedstawiony w kolejnym rozdziale został szczegółowo opisany w artykule [1]. W niniejszej pracy skupiono się raczej na analizie numerycznych rozwiązań modelu i ocenie ich adekwatności w odniesieniu do opisanej wyżej strategii wyszukiwania kryjówek i dyspersyjno-dyfuzyjnego charakteru rozwoju pojedynczej kolonii. Analizy rozwiązań dokonano przy użyciu interaktywnego arkusza programu Maple 16. Dużą część pracy stanowi opis konstrukcji samego arkusza jego użycia. Arkusz dostępny jest online i pozwala na weryfikację zaprezentowanych tutaj wyników.

4.1. Model matematyczny

Model matematyczny opisujący rozwój populacji nietoperzy i ich zachowanie oparty jest na układzie n równań różniczkowych, gdzie $n \in \mathbb{N}$ jest liczbą kryjówek w rozpatrywanym terenie.

$$x_i'(t) = \underbrace{k_i(x_i(t))x_i(t)}_{(1)} + \underbrace{\left[\sum_{j=1, j \neq i}^n \varphi(\xi_{ij}) (k_0 - k_j(x_j(t))) \cdot x_j(t) \right]}_{(2)} \cdot \underbrace{J\left(\frac{x_i(t)}{c_i}\right)}_{(3)} \quad (2.1)$$

dla $i = 1, 2, \dots, n$. Poniżej zdefiniowano zmienne, funkcje i parametry oraz podano interpretację poszczególnych części modelu.

Zmienna x_i , $i = 1, 2, \dots, n$ opisuje wielkość populacji nietoperzy w i -tej kryjówce. W modelu użyto następujących współczynników i funkcji:

- k_0 współczynnik narodzin, wspólny dla całej populacji nietoperzy;
- c_i optymalna pojemność kryjówki (dziupli). Określenie „optymalna pojemność” oznacza nie tylko wielkość kryjówki, ale także opisuje inne jej zalety, takie jak dogodny mikroklimat czy też właściwe położenie zapewniające ochronę przed drapieżnikami lub niekorzystnymi warunkami pogodowymi, itp.;
- $k_i : [0, \infty) \rightarrow (-k_0, k_0)$, $i = 1, 2, \dots, n$ funkcja wzrostu zależna od wielkości populacji w i -tej kryjówce. Funkcja opisuje przyrost i ubytek populacji, składa się nań, rozmnażanie, śmiertelność oraz migracja. Zakładamy, że funkcja k_i jest ciągła

- na przedziale $(c_i - \eta, c_i + \eta)$, gdzie η jest dowolnie małe, równa k_0 dla argumentów z przedziału $(0, c_i - \eta)$ oraz równa $-k_0$ dla argumentów z przedziału $(c_i + \eta, \infty)$;
- ξ_{ij} odległość pomiędzy i -tą a j -tą kryjówką;
 - a maksymalny zasięg detekcji nietoperzy przy poszukiwaniu kryjówek do zasiedlenia;
 - $\varphi : (0, \infty) \rightarrow (0, 1)$ funkcja odległości pomiędzy dwiema dowolnymi kryjówkami. Zakładamy, że φ jest malejąca na przedziale $(0, a)$ oraz równa 0 poza nim;
 - J funkcja odzwierciedla punkt 8) strategii opisaną w rozdziale 1, czyli ewentualną decyzję o migracji części osobników oraz kontynuację dalszych poszukiwań pozostałej części kolonii pozostającej w starym siedlisku. Zakładamy, że funkcja J jest różniczkowalna na swojej dziedzinie oraz równa 1 dla argumentów z przedziału $(0, 1 - \sigma)$, gdzie σ jest dowolnie małe oraz równa 0 dla argumentów z przedziału $(1, \infty)$.

Części modelu oznaczone numerami od (1) do (3) opisują poszczególne etapy rozwoju kolonii nietoperzy oraz jej dyspersję na podpopulacje zamieszkujące osobne kryjówki.

1. W początkowej fazie rozwoju kolonia nietoperzy zamieszkująca i -tą kryjówkę korzysta jedynie z jej zasobów, czyli wykorzystuje jej optymalną pojemność oraz korzysta ze źródeł pożywienia znajdujących się w otoczeniu i -tej kryjówki.
2. Gdy pojemność i -tej kryjówki zaczyna być niewystarczająca dochodzi do migracji części osobników. Rozpoczyna się dyspersja kolonii do kryjówek znajdujących się w obrębie maksymalnej detekcji. Oznacza to, że część kolonii przenosi się do nowych kryjówek położonych w odległości nie większej niż a od zamieszkiwanej i -tej kryjówki.
3. Ta część modelu opisuje „atrakcyjność” i -tej kryjówki, która spada wraz ze wzrostem liczebności populacji ją zamieszkującej. Użycie funkcji J opisuje ewentualną decyzję o migracji części osobników kolonii i kontynuacji dalszych poszukiwań nowych, dogodnych kryjówek przez osobniki pozostające w starym i -tym siedlisku.

W pracy [1] udowodniono, że poszczególne rozwiązania x_i , $i = 1, 2, \dots, n$ modelu (2.1) stabilizują się na poziomie bliskim wartości c_i , czyli optymalnej pojemności i -tej dziupli.

4.2. Interaktywny arkusz Maple

Przedstawimy tutaj dokładną konstrukcję interaktywnego arkusza Maple 16, który pozwala analizować numeryczne rozwiązania modelu (2.1). Maple to program komputerowy typu CAS służący do wykonywania obliczeń symbolicznych. Język programowania Maple jest językiem interpretowanym o dynamicznych typach danych. Obsługiwane są w nim obliczenia symboliczne i numeryczne oraz wizualizacja. Program pozwala także tworzyć niestandardowe interfejsy użytkownika, w tym

interaktywne arkusze, które pozwalają na analizowanie kilku procesów lub wątków jednocześnie. Arkusz interaktywny złożony jest z programowalnych, wbudowanych komponentów (*Embedded Components*) programu Maple. Komponenty to proste elementy interfejsu graficznego, które można osadzić w dokumencie, a następnie połączyć ze sobą odpowiednią programowalną akcją. Tworząc arkusz pozwalający na dynamiczną obserwację zmian zachowań rozwiązań numerycznych modelu (2.1) przy dynamicznej zmianie parametrów użyliśmy komponentów trzech typów: *Etykieta* (*Label*), *Suwak* (*Slider*) oraz *Wykres* (*Plot*). Każdy z komponentów został zaprogramowany w jednolity sposób, opisany w ramce Listing 1. Pierwsza część arkusza (rysunek 4.2) to tabela „Współrzędne kryjówek”. Posługując się suwakami użytkownik może określić położenie kryjówek w terenie. W aplikacji rozważamy cztery kryjówki A, B, C oraz D, które traktowane są jako cztery punkty położone w wycinku płaszczyzny kartezjańskiej $\mathbb{R}^2$ o wymiarach 200×200 . W części „Odległość między kryjówkami” automatycznie wyliczane są odległości między nimi. Co pozwala ustalić czy są one w promieniu maksymalnej detekcji nietoperzy czy też ich oddalenie utrudnia do nich dostęp, a co za tym idzie ich zasiedlenie. Odległość między kryjówkami wyliczana jest przy użyciu standardowego wzoru na odległość dwóch punktów na płaszczyźnie kartezjańskiej $\mathbb{R}^2$. Przy zmianie współrzędnych kryjówek pojawia się dynamiczna wizualizacja rozmieszczenia kryjówek w terenie (wykres „Układ kryjówek w terenie”). Używając suwaków w części „Warunki początkowe” użytkownik może określić początkowy (w chwili $t = 0$) stopień zasiedlenia poszczególnych dziupli drzew. Wielkości $x_1(0)$, $x_2(0)$, $x_3(0)$ oraz $x_4(0)$ odpowiadają w podanej kolejności początkowemu zasiedleniu kryjówek drzew A, B, C oraz D. Podobnie sterować też można optymalną pojemnością poszczególnych kryjówek zmieniając współczynniki c_1 , c_2 , c_3 oraz c_4 , które w podanej kolejności odpowiadają kryjówkom A, B, C oraz D. Możliwe jest także ręczne ustawienie współczynnika maksymalnego zasięgu detekcji nietoperzy. Arkusz pozwala na wybór współczynnika w zakresie od 5 do 90 metrów, w zależności czy rozważamy odpowiednio teren trudny, pełen przeszkód czy też przeciwnie, jest to obszar wolny od obiektów, które mogłyby zakłócać wyszukiwanie kryjówek. Sterować też możemy w niewielkim zakresie współczynnikiem narodzin wybierając wartość z przedziału $(0, 1)$. Nietoperze odznaczają się niskim poziomem rozrodu, wydając na świat zwykle tylko jedno młode stąd niewielki zakres wartości tego współczynnika. Pozostałe części arkusza to między innymi wykresy poszczególnych funkcji – składowych modelu, czyli funkcji φ , J oraz k_i dla $i = 1, 2, 3, 4$. Kluczowe dla analizy numerycznej modelu są wykresy jego rozwiązań umieszczone w centralnej części arkusza. Rysunek 4.2 prezentuje wygląd całego arkusza interaktywnego. Każde użycie dowolnego *Suwaka* powoduje dynamiczne zmiany wartości wyświetlane w odpowiadających *Etykietach* oraz zmiany na poszczególnych *Wykresach*.

Programując poszczególne komponenty arkusza użyliśmy następujących funkcji

$$\varphi(x) = \begin{cases} 1 - \frac{x}{a} & \text{dla } x < a, \\ 0 & \text{dla } x \geq a \end{cases},$$

$$J(x) = \begin{cases} 1 & \text{dla } x < \frac{7}{8} \\ \frac{1}{2} + \frac{1}{2} \sin(-8\pi x + \frac{15}{2}\pi) & \text{dla } \frac{7}{8} \leq x < 1 \\ 0 & \text{dla } x \geq 1 \end{cases}$$

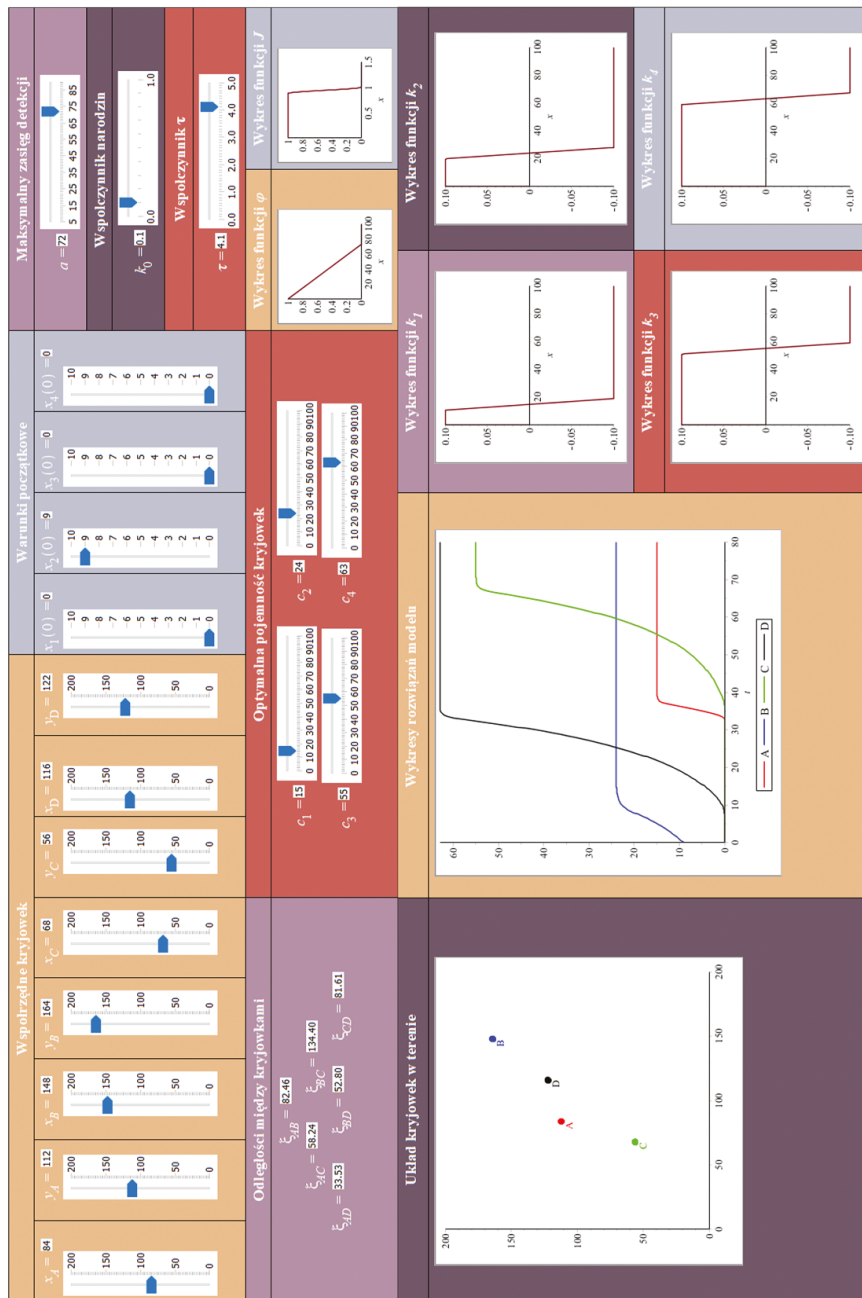
oraz

$$k_i(x) = \begin{cases} k_0 & \text{dla } x < c_i - \tau \\ \frac{k_0}{\tau}(c_i - x) & \text{dla } c_i - \tau \leq x < c_i + \tau \\ -k_0 & \text{dla } x \geq c_i + \tau \end{cases},$$

gdzie $i = 1, 2, 3, 4$. Arkusz pozwala na sterowanie wartością współczynnika τ w zakresie od 0 do 5. Tak dobrane funkcje spełniają opisane wcześniej założenia. W szczególności funkcja φ jest malejąca w przedziale $(0, a)$, a funkcja J spełnia założenie o różniczkowalności na całej swojej dziedzinie. Dodatkowo przyjęta liniowość funkcji φ na przedziale $(0, a)$ obrazuje proporcjonalny spadek zainteresowania nietoperzy kryjówkami położonymi coraz dalej od ich pierwotnego siedliska. Podobnie przyjęta ścisła monotoniczność funkcji J na przedziale $(1 - \sigma, 1)$ dla $\sigma = \frac{1}{8}$ obrazuje stopniowy spadek „atrakcyjności” kryjówki wraz ze wzrostem liczebności populacji ją zamieszkującej.

Firma MapleSoft rozwijająca i dystrybuująca program Maple udostępnia serwer MapleCloud, na którym użytkownicy mogą zamieszczać swoje arkusze oraz udostępniać je całej społeczności lub wybranym osobom. Aktywne korzystanie z udostępnionych arkuszy nie wymaga od użytkownika posiadania programu Maple. Interaktywny arkusz opisany w pracy znajduje się pod adresem: <https://maple.cloud/app/5328086670770176/Rozwiazania+modelu+opisujacego+rozwoj+podpopulacji+nietoperzy>. Nazwa dokumentu: *Rozwiązania modelu opisującego rozwój podpopulacji nietoperzy*. Data publikacji: 10.01.2021.

Niestety, część arkuszy interaktywnych udostępnianych online nie działa do końca prawidłowo lub wizualnie odbiega od pierwotnej wersji. Wynika to z pewnych niedociągnięć po stronie producenta. Tak jest też w przypadku opisywanego arkusza. Jego wygląd (krój i wielkość czcionki, kolorystyka itp.) odbiega nieco od tego co zamieszczono na rysunku 4.2. Udostępniony arkusz można też pobrać na swój komputer i korzystać z niego bezpośrednio w aplikacji Maple bez opisanych wyżej problemów.



RYSUNEK 4.2. Wygląd arkusza interaktywnego

ŹRÓDŁO: opracowanie własne

LISTING 4.6. Kod programujący każdy komponent interaktywnego dokumentu

ŹRÓDŁO: opracowanie własne

use DocumentTools, plots, plottools in

```
#Coordinates*of*tree*cavities
```

```
Do(xA = %Slider1);
```

```
Do(%Label1(caption) = sprintf("%.0f", xA));
```

```
Do(yA = %Slider2);
```

```
Do(%Label2(caption) = sprintf("%.0f", yA));
```

```
Do(xB = %Slider3);
```

```
Do(%Label3(caption) = sprintf("%.0f", xB));
```

```
Do(yB = %Slider4);
```

```
Do(%Label4(caption) = sprintf("%.0f", yB));
```

```
Do(xC = %Slider5);
```

```
Do(%Label5(caption) = sprintf("%.0f", xC));
```

```
Do(yC = %Slider6);
```

```
Do(%Label6(caption) = sprintf("%.0f", yC));
```

```
Do(xD = %Slider7);
```

```
Do(%Label7(caption) = sprintf("%.0f", xD));
```

```
Do(yD = %Slider8);
```

```
Do(%Label8(caption) = sprintf("%.0f", yD));
```

```
#Initial*conditions
```

```
Do(x10 = %Slider9);
```

```
Do(%Label9(caption) = sprintf("%.0f", x10));
```

```
Do(x20 = %Slider10);
```

```
Do(%Label10(caption) = sprintf("%.0f", x20));
```

```
Do(x30 = %Slider11);
```

```
Do(%Label11(caption) = sprintf("%.0f", x30));
```

```

Do(x40 = %Slider12);
Do(%Label12(caption) = sprintf("%.0f", x40));

#Distances*between*the*tree*cavities
Do(d12=sqrt((%Slider1-%Slider3)^2+(%Slider2-%Slider4)^2));
Do(%Label17(caption) = sprintf("%.2f", d12));

Do(d13=sqrt((%Slider1-%Slider5)^2+(%Slider2-%Slider6)^2));
Do(%Label18(caption) = sprintf("%.2f", d13));

Do(d23=sqrt((%Slider3-%Slider5)^2+(%Slider4-%Slider6)^2));
Do(%Label19(caption) = sprintf("%.2f", d23));

Do(d14=sqrt((%Slider1-%Slider7)^2+(%Slider2-%Slider8)^2));
Do(%Label20(caption) = sprintf("%.2f", d14));

Do(d24=sqrt((%Slider3-%Slider7)^2+(%Slider4-%Slider8)^2));
Do(%Label21(caption) = sprintf("%.2f", d24));

Do(d34=sqrt((%Slider5-%Slider7)^2+(%Slider6-%Slider8)^2));
Do(%Label22(caption) = sprintf("%.2f", d34));

#Location*of*the*tree*cavities
Do(A=textplot([%Slider1-2,%Slider2-2, 'typeset'("A ")], 'align'='below', color=red));

Do(B=textplot([%Slider3-2,%Slider4-2, 'typeset'("B ")], 'align'='below', color=blue));

Do(C=textplot([%Slider5-2,%Slider6-2, 'typeset'("C ")], 'align'='below', color=green));

Do(F=textplot([%Slider7-2,%Slider8-2, 'typeset'("D ")], 'align'='below'));

Do(%Plot1=plots[display]([A,circle([%Slider1,%Slider2], 1,color=red),B,circle([%Slider3,%Slider4], 1,color=blue),C,circle([%Slider5,%Slider6], 1,color=green),F,circle([%Slider7,%Slider8], 1,color=black)], view = [0 .. 200, 0 .. 200], thickness = 5));

```

```

#Optimal*capacities*of*the*tree*cavities
Do(c1 = %Slider23);
Do(%Label23(caption) = sprintf("%.0f", c1));

Do(c2 = %Slider24);
Do(%Label24(caption) = sprintf("%.0f", c2));

Do(c3 = %Slider25);
Do(%Label25(caption) = sprintf("%.0f", c3));

Do(c4 = %Slider26);
Do(%Label26(caption) = sprintf("%.0f", c4));
#Maximum*range*of*the*detection
Do(a = %Slider31);
Do(%Label31(caption) = sprintf("%.0f", a));

#Birth*coefficient
Do(k0 = %Slider32);
Do(%Label32(caption) = sprintf("%.1f", k0));

#Coefficient*tau
Do(tau = %Slider33);
Do(%Label33(caption) = sprintf("%.1f", tau));

#Graph*of*the*function*k_1
Do(k1 = proc (x) options operator, arrow; piecewise(x <= 0, 0, x <= c1-tau, k0, x <= c1+tau,
-k0*x/tau+k0*c1/tau, c1+tau < x, -k0) end proc);
Do(%Plot4 = plot(k1(x), x = 0 .. 100));

#Graph*of*the*function*k_2
Do(k2 = proc (x) options operator, arrow; piecewise(x <= 0, 0, x <= c2-tau, k0, x <= c2+tau,
-k0*x/tau+k0*c2/tau, c2+tau < x, -k0) end proc);
Do(%Plot5 = plot(k2(x), x = 0 .. 100));

#Graph*of*the*function*k_3
Do(k3 = proc (x) options operator, arrow; piecewise(x <= 0, 0, x <= c3-tau, k0, x <= c3+tau,
-k0*x/tau+k0*c3/tau, c3+tau < x, -k0) end proc);
Do(%Plot6 = plot(k3(x), x = 0 .. 100));

```

```
#Graph*of*the*function*k_4
```

```
Do(k4 = proc (x) options operator, arrow; piecewise(x <= 0, 0, x <= c4-tau, k0, x <= c4+tau,  
-k0*x/tau+k0*c4/tau, c4+tau < x, -k0) end proc);
```

```
Do(%Plot7 = plot(k4(x), x = 0 .. 100));
```

```
#Graph*of*the*function*phi
```

```
Do(phi = proc (x) options operator, arrow; piecewise(x < 0, 0, x <= a, -x/a+1, a < x, 0) end proc);
```

```
Do(%Plot8 = plot(phi(x), x = 0 .. 100));
```

```
#Graph*of*the*function*J
```

```
Do(J = proc (x) options operator, arrow; piecewise(x < 7/8, 1, 7/8 <= x and x < 1, 1/2+(1/2)*sin(-  
8*Pi*x+(15/2)*Pi), 7/8 <= x, 0) end proc);
```

```
Do(%Plot9 = plot(J(x), x = 0 .. 1.5));
```

```
#System*first*part
```

```
Do(row1 = (diff(x1(t), t) = k1(x1(t))*x1(t)+(phi(d12)*(k0-k2(x2(t))))*x2(t)+phi(d13)*(k0-  
-k3(x3(t)))*x3(t)+phi(d14)*(k0-k4(x4(t)))*x4(t))*J(x1(t)/  
c1)));
```

```
Do(row2 = (diff(x2(t), t) = k2(x2(t))*x2(t)+(phi(d12)*(k0-k1(x1(t)))*x1(t)+phi(d23)*(k0-k3(x3(t)))*  
x3(t)+phi(d24)*(k0-k4(x4(t)))*x4(t))*J(x2(t)/c2)));
```

```
Do(row3 = (diff(x3(t), t) = k3(x3(t))*x3(t)+(phi(d13)*(k0-k1(x1(t)))*x1(t)+phi(d23)*(k0-k2(x2(t)))*  
x2(t)+phi(d34)*(k0-k4(x4(t)))*x4(t))*J(x3(t)/c3)));
```

```
Do(row4 = (diff(x4(t), t) = k4(x4(t))*x4(t)+(phi(d14)*(k0-k1(x1(t)))*x1(t)+phi(d24)*(k0-k2(x2(t)))*  
x2(t)+phi(d34)*(k0-k3(x3(t)))*x3(t))*J(x4(t)/c4)));
```

```
Do(sys = dsolve({row1, row2, row3, row4, x1(0) = x10, x2(0) = x20, x3(0) = x30, x4(0) = x40},  
{x1(t), x2(t), x3(t), x4(t)}, type = numeric, output = listprocedure));
```

```
Do(%Plot2 = odeplot(sys, [[t, x1(t), color = red], [t, x2(t), color = blue], [t, x3(t), color = green],  
[t, x4(t), color = black]], 0 .. 80, legend=[„A“, „B“, „C“, „D“]));
```

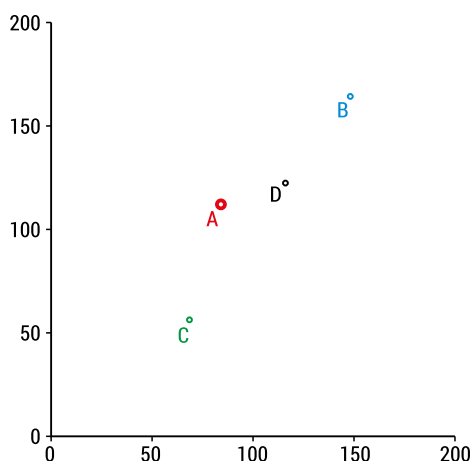
```
end use;
```

4.3. Analiza wybranych przypadków przy użyciu arkusza interaktywnego Maple

W tym rozdziale rozważy się pięć przypadków przy różnym układzie kryjówek w terenie, różnym początkowym ich zasiedleniu, zakładanych pojemnościach optymalnych oraz przy różnej maksymalnej detekcji opisującej pośrednio odmienny, pod względem liczby przeszkód, rozważany teren.

Przypadek 1

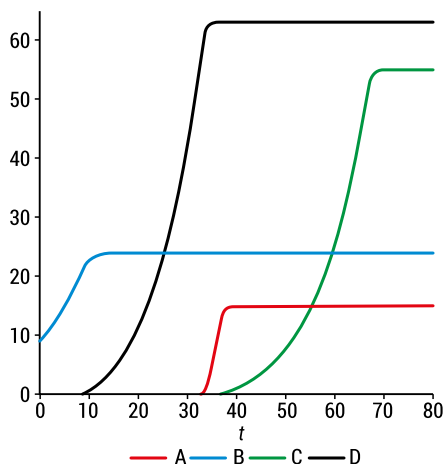
Rozważa się cztery kryjówki o współrzędnych $A = (84, 112)$, $B = (148, 164)$, $C = (68, 56)$, $D = (116, 122)$ (rysunek 4.3).



RYSUNEK 4.3. Układ kryjówek w terenie. Przypadek 1

ŹRÓDŁO: opracowanie własne

W chwili $t = 0$ zasiedlona jest jedynie kryjówka B , gdzie $x_2(0) = 9$ oraz $x_1(0) = x_2(0) = x_3(0) = 0$. Optymalna pojemność zasiedlonej dziupli jest równa $c_2 = 24$. Odległości zamieszkałej kryjówki B od pozostałych dziupli drzew są następujące $\xi_{AB} = 82,46$, $\xi_{BC} = 134,4$, $\xi_{BD} = 52,8$, (ustawienia arkusza interaktywnego na rysunku 4.2). Przy zasięgu detekcji ustawionym na poziomie $a = 72$ jedynie dziupla D pozostaje w zasięgu detekcji kolonii. Jak widać na rysunku 4.4, w początkowej fazie rozwoju kolonia nietoperzy zamieszkująca kryjówkę B korzysta jedynie z jej zasobów. W chwili, gdy liczebność populacji zaczyna być bliska optymalnej pojemności dziupli rozpoczyna się migracja, najpierw do kryjówki D jedynej dostępnej w obrębie detekcji, po wyczerpaniu i jej zasobów następuje dyspersja kolonii do kolejnych kryjówek A i C .

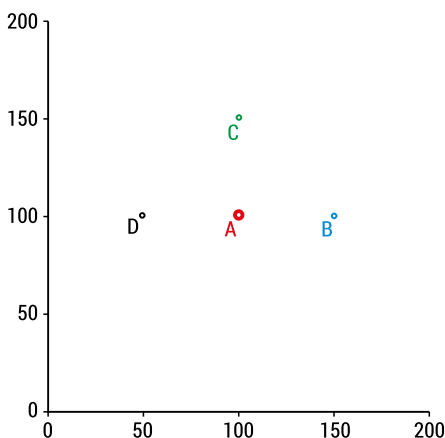


RYSUNEK 4.4. Wykresy rozwiązań modelu. Przypadek 1

ŹRÓDŁO: opracowanie własne

Przypadek 2

Rozważa się cztery kryjówki o współrzędnych $A = (100, 100)$, $B = (150, 100)$, $C = (100, 150)$, $D = (50, 100)$ (rysunek 4.5).

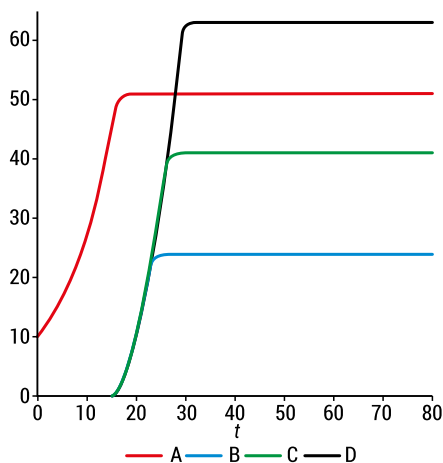


RYSUNEK 4.5. Układ kryjówek w terenie. Przypadek 2

ŹRÓDŁO: opracowanie własne

W chwili początkowej zasiedlona jest jedynie kryjówka A , gdzie $x_1(0) = 10$. Pozostałe trzy kryjówki B , C i D położone są w jednakowych odległościach od zamieszkałej dziupli A . Rysunek 4.6 prezentuje wykresy rozwiązań przy zasięgu detekcji $a = 89$, czyli w przypadku obszaru wolnego od obiektów zakłócających wyszukiwanie kryjówek. Jak widać po wyczerpaniu zasobów zamieszkałej kryjówki następuje migracja

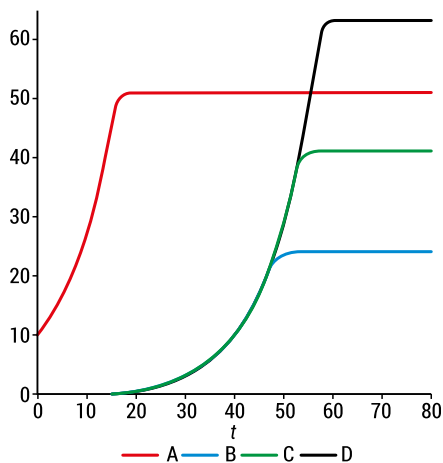
i jednoczesna dyspersja do wszystkich pozostałych dostępnych kryjówek równooddalonych od dziupli A. Arkusz interaktywny w opisywanym przypadku prezentuje rysunek 4.13.



RYSUNEK 4.6. Wykresy rozwiązań modelu. Przypadek 2 dla $a = 89$

ŹRÓDŁO: opracowanie własne

Gdy przy tym samym układzie kryjówek rozważy się jednak teren trudny, pełen przeszkód ($a = 51$) utrudniających poszukiwania następuje powolniejsza migracja, rozciągnięta w czasie (rysunki 4.7 oraz 4.14).

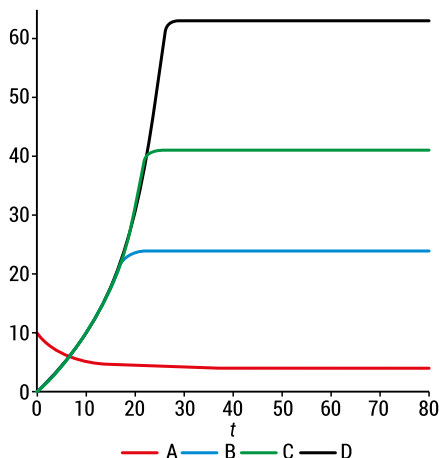


RYSUNEK 4.7. Wykresy rozwiązań modelu. Przypadek 2 dla $a = 51$

ŹRÓDŁO: opracowanie własne

Przypadek 3

Rozpatruje się ten sam układ kryjówek w terenie jak w Przypadku 2. I podobnie jak wyżej zasiedlona jest jedynie kryjówka **A**. Tym razem jednak początkowa liczebność kolonii nietoperzy zamieszkująca kryjówkę **A** przekracza optymalną pojemność dziupli. Mamy $x_1(0) = 10$ przy $c_1 = 4$ (rysunek 4.15). Następuje natychmiastowa migracja do wszystkich dostępnych wolnych kryjówek lub też wymieranie populacji do poziomu bliskiego optymalnej pojemności kryjówki – rysunek 4.8.

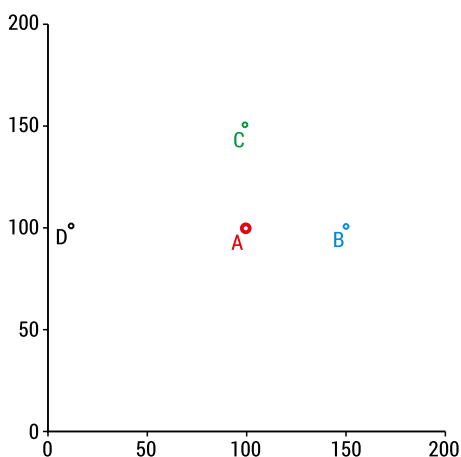


RYSUNEK 4.8. Wykresy rozwiązań modelu. Przypadek 3

ŹRÓDŁO: opracowanie własne

Przypadek 4

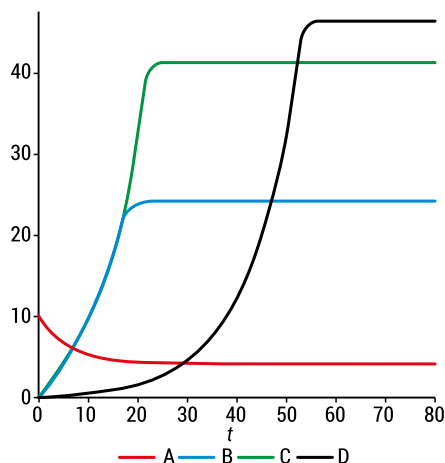
Zmienia się położenie kryjówki **D** względem jej pierwotnego umiejscowienia opisanego w Przypadku 3 (rysunek 4.9).



RYSUNEK 4.9. Układ kryjówek w terenie. Przypadek 4

ŹRÓDŁO: opracowanie własne

Tym razem $D = (12, 100)$ oraz $\xi_{AD} = 88$. Pozostałe ustawienia arkusza pozostają bez zmian, czyli $x_1(0) = 10$, przy $c_1 = 4$, $x_2(0) = x_3(0) = x_4(0) = 0$, $a = 90$. Wszystkie kryjówki są w zasięgu detekcji kolonii nietoperzy zamieszkujących kryjówkę A . Jednak migracja w pierwszej kolejności następuje do kryjówek najbliższych położonych B i C , a dopiero potem do kryjówki D (rysunki 10 i 16).

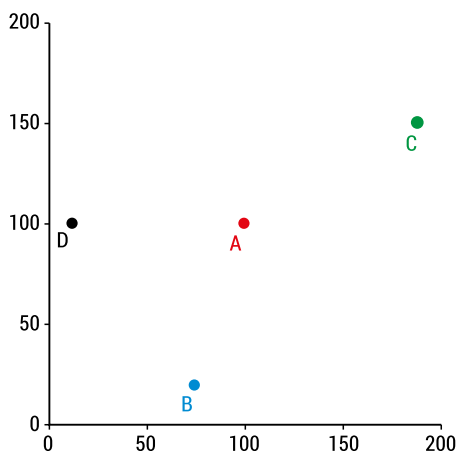


RYSUNEK 4.10. Wykresy rozwiązań modelu. Przypadek 4

ŹRÓDŁO: opracowanie własne

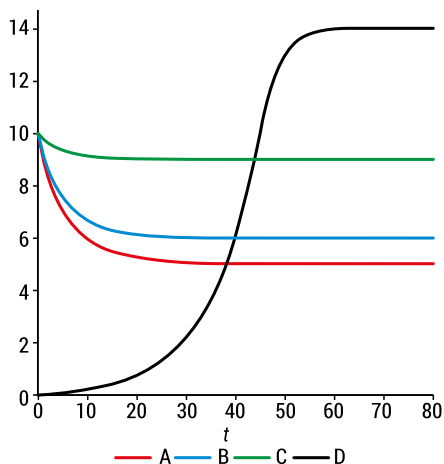
Przypadek 5

Rozpatruje się trzy zasiedlone i przepełnione kryjówki A , B i C , gdzie $x_1(0) = x_2(0) = x_3(0) = 10$ przy $c_1 = 5$, $c_2 = 6$, $c_3 = 9$ w układzie jak na rysunku 4.11.



RYSUNEK 4.11. Układ kryjówek w terenie. Przypadek 5

ŹRÓDŁO: opracowanie własne



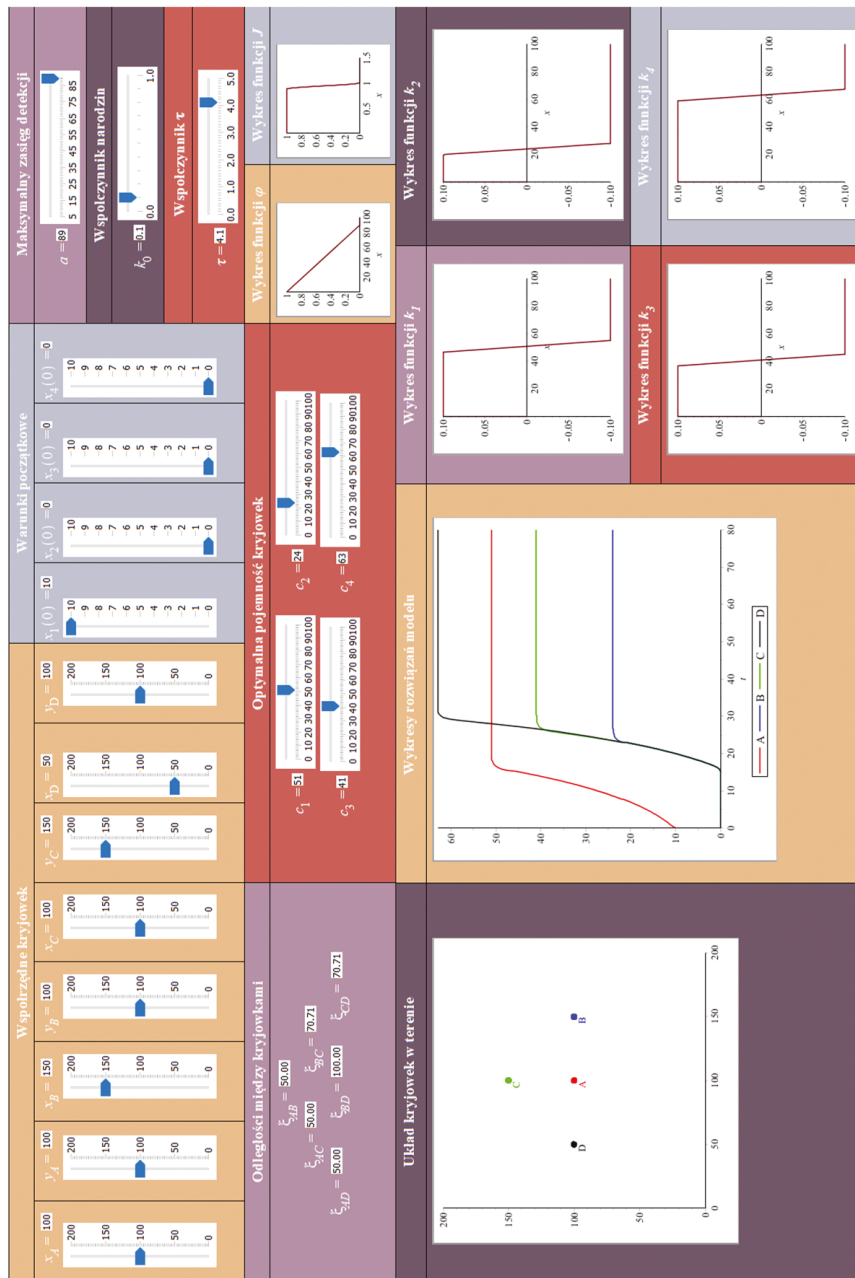
RYSUNEK 4.12. Wykresy rozwiązań modelu. Przypadek 5

ŹRÓDŁO: opracowanie własne

Jedyną wolną kryjówką gotową do zamieszkania jest dziupla **D** ($x_4(0) = 0$) znajdująca się w obrębie detekcji osobników zamieszkujących dziuplę **A**. Przyjęliśmy $a = 89$ oraz $\xi_{AD} = 88$, $\xi_{BD} = 101,21$, $\xi_{CD} = 182,96$ (rysunek 4.17). W tym przypadku następuje natychmiastowy spadek liczebności podpopulacji w przepelnionych kryjówkach **A**, **B** i **C** z powodu śmiertelności lub migracji do pustej kryjówki **D**. Analizując wykresy rozwiązań prezentowanych na rysunku 4.12 trudno wnioskować o migracji z najbliższej położonej dziupli **A**. Natomiast prezentowane wcześniejsze cztery przypadki wskazują na poprawną konstrukcję modelu, który symuluje strategię wyszukiwania kryjówek adekwatnie do dobranej topografii terenu, czyli adekwatnie do ustalonego poziomu maksymalnej detekcji.

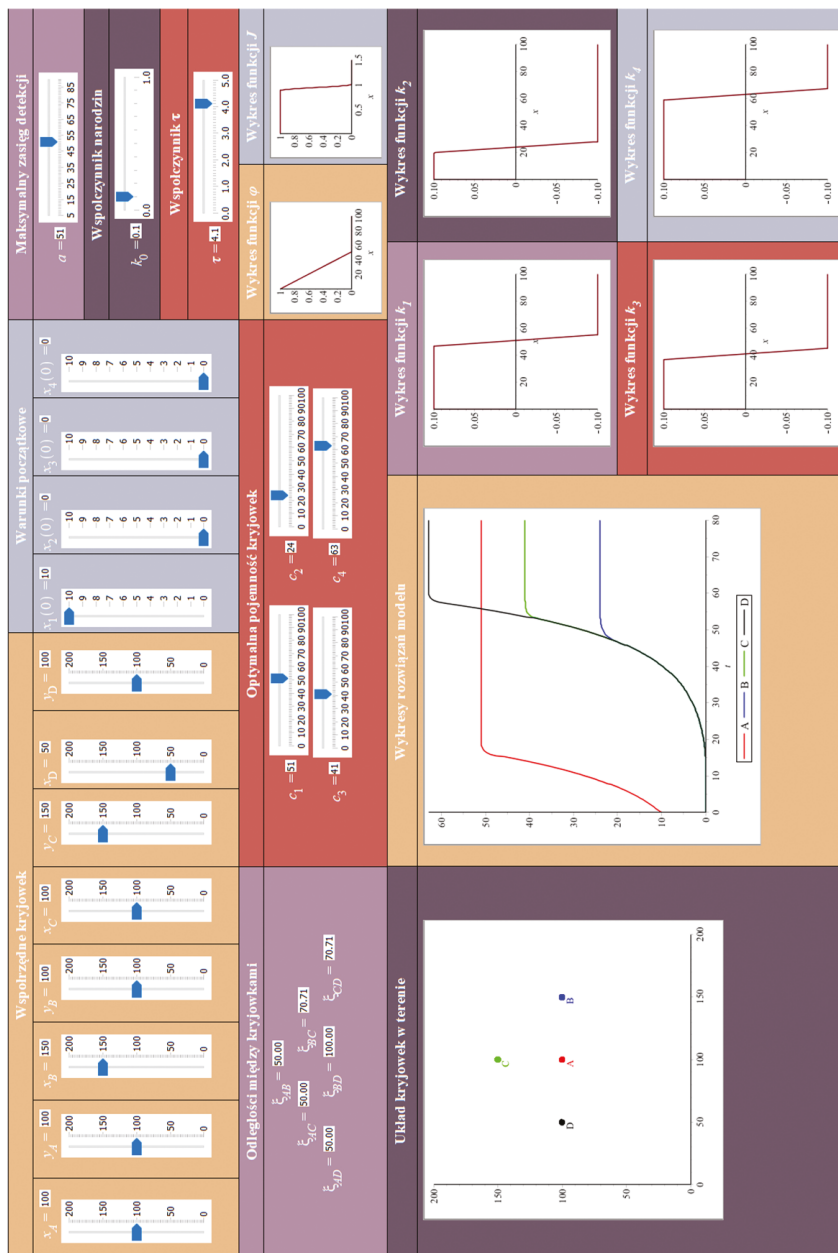
Podsumowanie

Konstrukcja modelu dobrze sprawdza się w opisanych wyżej przypadkach. Prawdłowo symuluje zwyczaję nietoperzy związane z komunikacją i przekazywaniem informacji między osobnikami jednej kolonii oraz wyszukiwaniem i zasiedlaniem naturalnych kryjówek w terenie. Prezentowany w pracy arkusz interaktywny pozwala czytelnikowi samodzielnie zweryfikować zaprezentowane w pracy wyniki oraz być może odkryć pewne niedociągnięcia i niedoskonałości w konstrukcji modelu teoretycznego.

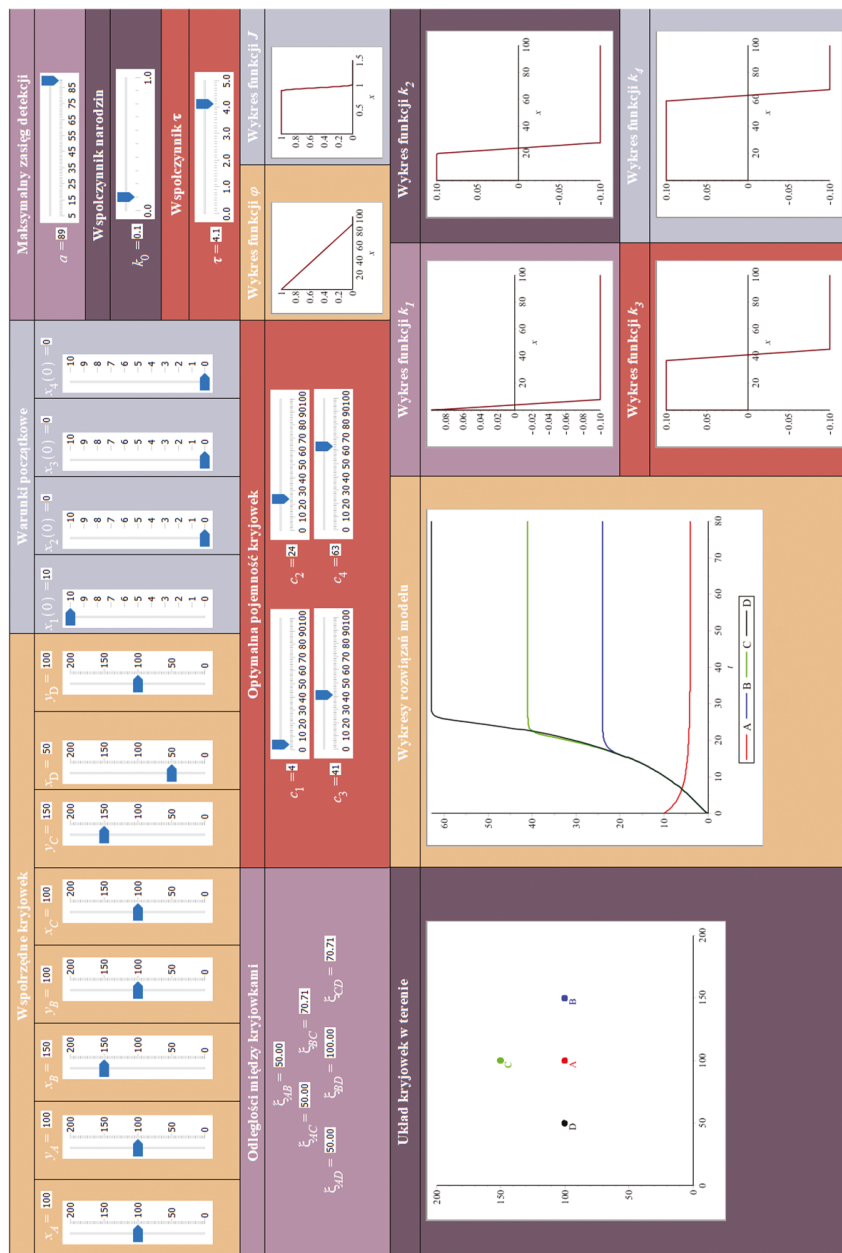


RYСУNEK 4.13. Ustawienia arkusza interaktywnego w Przypadku 2 dla $a = 89$

ŹRÓDŁO: opracowanie własne

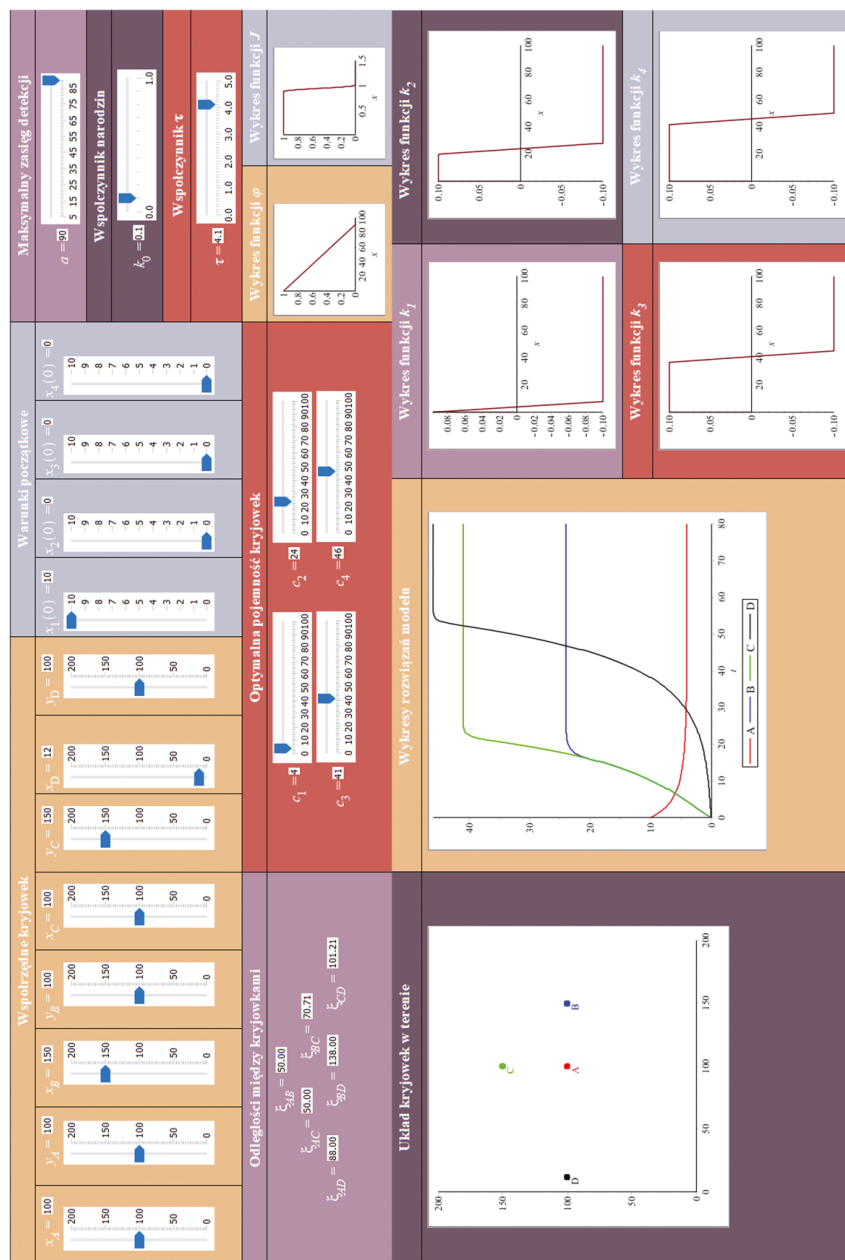
RYSUNEK 4.14. Ustawienia arkusza interaktywnego w Przypadku 2 dla $a = 51$

ŹRÓDŁO: opracowanie własne



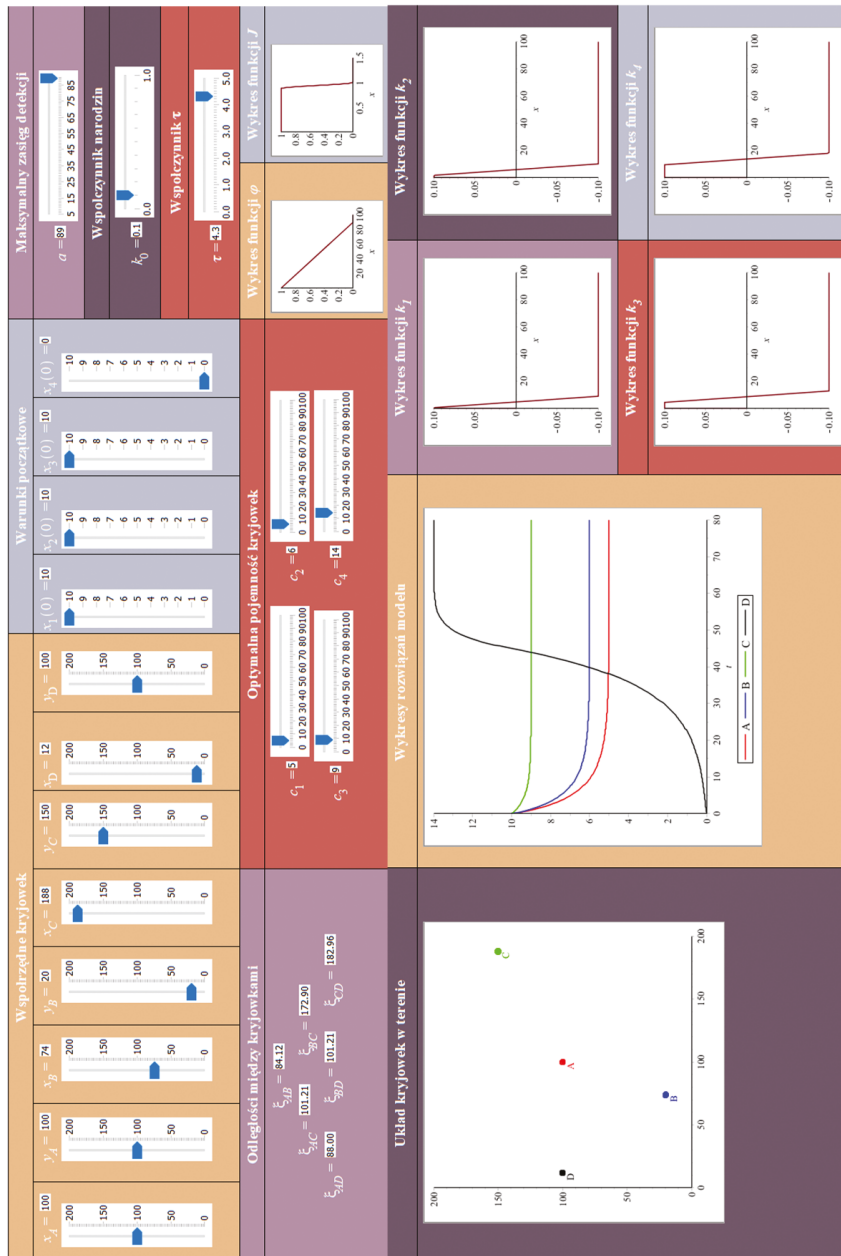
RYSUNEK 4.15. Ustawienia arkusza interaktywnego w Przypadku 3

ŹRÓDŁO: opracowanie własne



RYSUNEK 4.16. Ustawienia arkusza interaktywnego w Przypadku 4

ŹRÓDŁO: opracowanie własne



RYSUNEK 4.17. Ustawienia arkusza interaktywnego w Przypadku 5

ŹRÓDŁO: opracowanie własne

Bibliografia

- [1] Dawidowicz A. L., Poskrobko A., *Mathematical model of bats' subpopulations development*, "Mathematica Applicanda", 2018, 46 (1), s. 49-57
- [2] Furmankiewicz J., Ruczyński I., Urban R., Jones G., *Social calls provide tree-dwelling bats with information about the location of conspecifics at roosts*, "Ethology", 2011, 117, s. 480-489
- [3] Kerth G., *Group decision-making in fission-fusion societies*, "Behavioural Processes", 2006, 84, s. 662-663
- [4] Ruczyński I., Bartoń K. A., *Modelling sensory limitation: The role of tree selection, memory and information transfer in bats' roost searching strategies*, "PLOS ONE", 2012, 7(9): e44897
- [5] Ruczyński I., Kalko E. K. V., Siemers B. M., *The sensory basis of roost finding in a forest bat, *Myotis myotis**, "Journal of Experimental Biology", 2007, 210, s. 3607-3615
- [6] Ruczyński I., Kalko E. K. V., Siemers B. M., *Calls in the forest: how bats find tree cavities*, "Ethology", 2009, 115, s. 167-177
- [7] Schmeidel E., Girejko E., *On mathematical model of bats' roost searching strategies*, "Wulfenia Journal", 2016, 23, s. 67-75
- [8] Tsoar A., Nathan R., Bartan Y., Vyssotski A., Dell'Omo G., Ulanovsky N., *Large-scale navigational map in a mammal*, "Proceedings of the National Academy of Sciences", 2011, 108(37), E718-E724

Abstract

Solutions of the model describing the development of bats' sub-populations – analysis using interactive document of the Maple program

In the work, we analyze the behavior of the solutions to the mathematical model describing bats' population development. We present in detail the assumptions of the model and the natural motivations behind its construction. We take into account, among other, the specific searching roosts strategy, limited detection range in a difficult environment full of obstacles and the ability to transmit information in a group. The behavior of the bats population and their searching roosts strategy described in the paper is based on the observation of the behavior of two bats' species inhabiting the Białowieża Forest. The presented model is based on the system of ordinary differential equations. In the work, we present some simulations based on the interactive Maple document. Such document allows to analyze the behavior of numerical solutions under dynamically changing model parameters. We describe in detail the construction and operation the document, which uses the programmable Maple Embedded Components.

Keywords: interactive Maple document, bats, mathematical model