

# Modelowanie ewolucji populacji ze specjacją

Asia Dąbrowska i Ania Szymik

16 kwietnia 2024

Celem naszego projektu, było stworzenie modelu ewolucji populacji uwzględniającego specjację gatunków, który pomógłby w zrozumieniu dynamiki adaptacji organizmów na skutek zmian w środowisku.

## 1 Założenia

Stworzony przez nas model umożliwia przeprowadzenie symulacji ewolucji przez *generations* pokoleń. Podczas pojedynczego pokolenia zachodzą następujące etapy: mutacja, selekcja osobników, reprodukcja, sprawdzenie warunku na pojemność środowiska oraz zmiana w środowisku.

Na początku każdego pokolenia zachodzi mutacja poszczególnych genotypów w populacji. Z prawdopodobieństwem *mutation\_prob* dla pojedynczego osobnika zajdzie mutacja losowo wybranego genu. Wartość, o którą się ta ekspresja zmieni, jest losowana z rozkładu normalnego o odchyleniu standardowym *mutation\_std*.

Proces selekcji przebiega następująco: dla każdego osobnika jest obliczana wartość jego dostosowania przy użyciu miary (odwrotnie proporcjonalnej do odległości od optimum):

$$\phi_{\alpha} = \exp\left(-\frac{\|o - \alpha\|}{2\sigma^2}\right)$$

gdzie:

$o \rightarrow$  genotyp osobnika,

$\alpha \rightarrow$  optymalny genotyp

$\sigma \rightarrow$  odchylenie standardowe dostosowania organizmów względem optymalnego genotypu

Następnie z populacji usuwane są niewystarczająco dostosowane osobniki (znajdujące się za daleko od optimum). W przypadku, gdy w środowisku jest już więcej niż jeden optymalny genotyp, rozważamy ten najbliższy osobnikowi (dla którego wartość dostosowania do środowiska jest największa).

Reprodukcja przebiega bezpłciowo, poprzez klonowanie – kopiowanie genotypu organizmu rodzicielskiego – oraz jest zależna od dostosowania osobnika. Organizm po rozmnożeniu się umiera (długość życia organizmów w naszym modelu wynosi jedno pokolenie).

Zakończenie pokolenia następuje w wyniku zmiany w środowisku, która powoduje przesunięcie optymalnych genotypów. Przed tym zdarzeniem, sprawdzamy, czy wielkość populacji nie przekracza pojemności siedliska i jeżeli tak jest, redukujemy populację do tej wielkości.

Parametry opisujące populację:

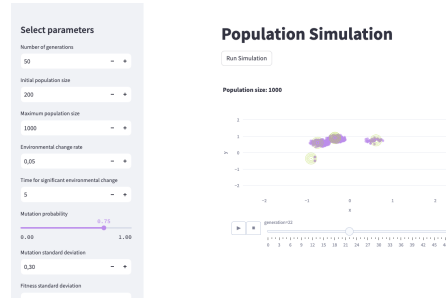
- $N \rightarrow$  liczebność populacji
- $n \rightarrow$  liczba genów pojedynczego osobnika
- *mutation\_prob*  $\rightarrow$  prawdopodobieństwo zajścia mutacji
- *mutation\_std*  $\rightarrow$  odchylenie standardowe rozkładu, z którego losujemy wartość zmiany ekspresji zmutowanego genu

Parametry opisujące środowisko:

- *max\_N*  $\rightarrow$  maksymalna pojemność środowiska
- *env\_change*  $\rightarrow$  stała opisująca ciągłą zmianę optymalnego genotypu
- $T \rightarrow$  okres czasu, po którym zachodzi specjacja - pojawia się nowy genotyp optymalny

- $angle \rightarrow$  kąt pomiędzy nowo powstałym optimum, a wektorem, z którego się ono wydzieliło
- $fitness\_thr \rightarrow$  minimalne dostosowanie osobnika umożliwiające mu przeżycie
- $fitness\_std \rightarrow$  odchylenie standardowe dostosowania genotypów organizmów względem optymalnego genotypu

## 2 Implementacja



Rysunek 1: Interfejs graficzny aplikacji

W naszej implementacji populacja reprezentowana jest jako macierz, której wiersze – wektory liczb – reprezentują genotypy poszczególnych osobników. Przyjęliśmy, że każdy osobnik charakteryzowany jest przez dwie liczby (geny), aby możliwa była łatwo interpretowalna wizualizacja, bez potrzeby redukcji wymiaru, oraz aby minimalizować czas renderowania symulacji. Osobno, również jako dwuelementowe wektory, ale tym razem w liście przechowywane są optymalne genotypy, których liczba zmienia się w trakcie symulacji.

Wszystkie dane, hiperparametry i metody przechowywane są w dedykowanej klasie, co ułatwia zarządzanie danymi symulacji, zapewniając centralizację i uporządkowanie informacji,

Przejdźmy zatem do opisu implementacji kolejnych kroków symulacji, powtarzających się w każdym pokoleniu – przebiegu pętli `for`:

- mutacja – dla każdego osobnika generowana jest liczba z rozkładu  $Uniform(0, 1)$  i jeśli jest ona mniejsza od prawdopodobieństwa mutacji ( $mutation\_prob$ ), losowany jest indeks, a następnie wartość z  $\mathcal{N}(0, mutation\_std^2)$  dodawana do wartości pod wylosowanym indeksem.
- selekcja – dla każdego osobnika obliczane jest maksimum wartości z funkcji fitness w stosunku do każdego z optymalnych genotypów, a następnie do macierzy reprezentującej populację przepisywane są tylko te osobniki, dla których wartość ta jest większa lub równa przyjętemu progowi  $fitness\_thr$ . Redukcję populacji do pojemności środowiska zaimplementowaliśmy jako losowe przetasowanie wierszy macierzy, a następnie zostawienie jedynie pierwszych  $max\_N$ .
- reprodukcja – przyjętym przez nas modelem reprodukcji jest klonowanie, a liczba pozostawionego potomstwa zależy od dostosowania osobnika. Zakres jego wartości od  $fitness\_th$  do 1 jest dzielony jednostajnie na  $max\_num\_children + 1$  przedziałów odpowiadających liczbie dzieci od 0 do  $max\_num\_children$  – może się zatem zdarzyć, że osobnik przeżyje, ale będzie za słabo dostosowany do środowiska, by pozostawić potomstwo. Dodatkowo, zgodnie z przyjętym założeniem, że każdy osobnik żyje tylko przez jedno pokolenie, na zmienną opisującą populację przepisywana jest nowopowstała macierz składająca się wyłącznie z osobników potomnych.
- zmiana środowiska – w każdym pokoleniu (poza tymi kiedy następuje duża zmiana) każdy z optymalnych genotypów zmienia się o  $env\_change$  jednostek w ustalonym przez swój wektor kierunku, zaś co  $T$  pokoleń następuje specjacja. Zaimplementowaliśmy ją jako rozejście się losowo wybranego optymalnego genotypu na dwa nowe genotypy poprzez jego rotację w dwóch przeciwnych kierunkach o kąt  $angle$ .

Aby nadać życie naszej symulacji i uczynić ją bardziej interaktywną, wykorzystaliśmy bibliotekę Plotly. Przy jej pomocy symulacja stała się animacją, którą można odtwarzać, zatrzymywać i

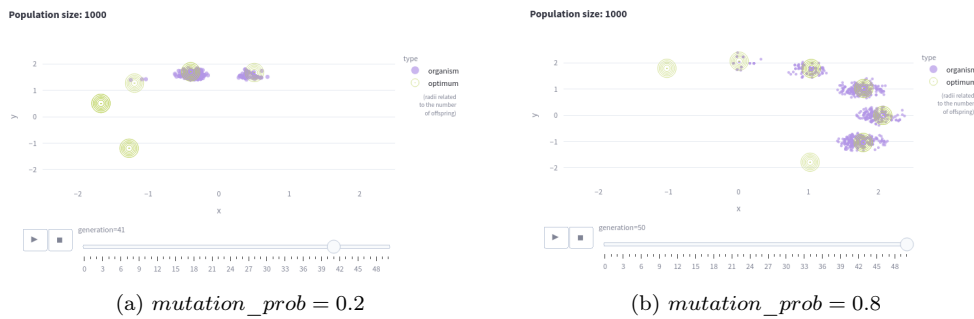
przewijać, a także, dzięki suwakowi pod wykresem, niezależnie analizować, co dzieje się z populacją w każdym pokoleniu.

Aby było to możliwe, konieczne jest przechowywanie danych o populacji i optymalnych genotypach w każdym pokoleniu. W tym celu wykorzystaliśmy obiekt `pandas.DataFrame`. Przechowujemy w nim, poza współrzędnymi punktów, także promienie okręgów reprezentujących odległość od optymalnego genotypu związaną z daną liczbą potomstwa.

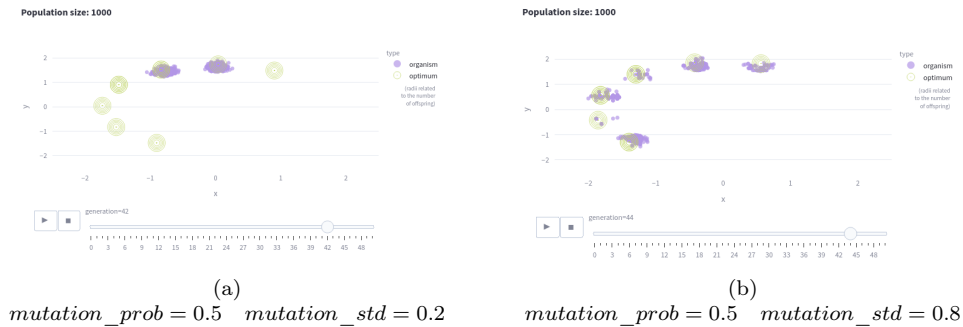
Całość opakowaliśmy w aplikację przy użyciu biblioteki Streamlit, dzięki czemu możliwe jest wygodne zmienianie parametrów symulacji i obserwowanie, jak w zależności od nich różnią się wyniki. Jej interfejs graficzny przedstawiony jest na Rys. 1.

### 3 Wyniki

Analizując symulację w zależności od kombinacji parametrów doszliśmy do kilku ciekawych obserwacji. Po pierwsze, nie zawsze przy nowopowstałych optymalnych genotypach mamy jakieś osobniki, a skorelowane jest to z prawdopodobieństwem zajścia mutacji oraz odchyleniem standardowym jej rozkładu (im te dwa parametry są mniejsze, tym częściej występuje taka sytuacja).



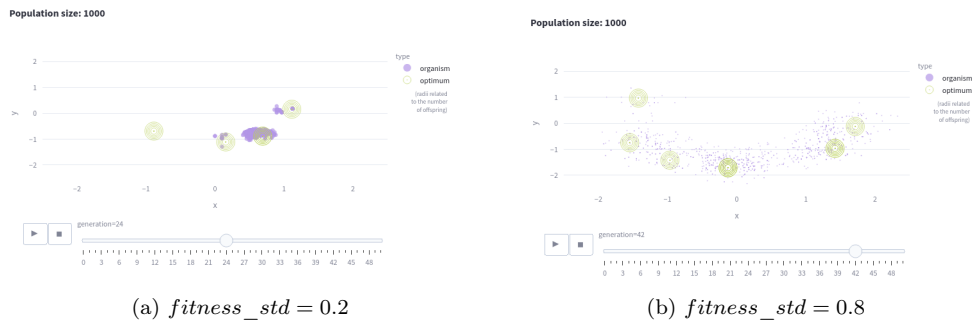
Rysunek 2: Wpływ  $mutation\_prob$  na ewolucję



Rysunek 3: Wpływ  $mutation\_std$  na ewolucję

Odkryliśmy też zależność między  $mutation\_prob$  a  $env\_change$  w kontekście liczebności populacji: jeśli prawdopodobieństwo zajścia mutacji jest za małe w stosunku do tego, jak szybko zmienia się środowisko, populacja wymiera – np. dla  $mutation\_prob = 0.3$  i  $env\_change = 0.1$ , przy pozostałych parametrach domyślnych, populacja w większości przypadków wymiera.

Kolejną ciekawą obserwacją jest wpływ  $fitness\_std$  na dynamikę populacji. Im mniejsza wartość tego parametru, tym genotypy osobników skupione są bliżej optimum. Natomiast  $fitness\_std \rightarrow \infty$  powoduje rozproszenie się genotypów i stworzenie się chmury punktów, zupełnie niezwiązanej z optimum.



Rysunek 4: Wpływ  $fitness\_std$  na ewolucję

Ponadto, zauważyliśmy, że jeśli redukcja liczebności populacji do  $max\_N$  następowała przy selekcji, przed reprodukcją, liczebność naszej populacji dążyła do  $max\_num\_children \cdot max\_N$ , dlatego zdecydowaliśmy się przenieść ją po reprodukcji.

Dodatkowo, dzięki temu, że najlepiej dostosowane osobniki pozostawiają najwięcej potomstwa, nawet przy  $fitness\_thr$  równym 0, osobniki i tak skupiają się wokół optymalnych genotypów, jeśli  $fitness\_std$  jest odpowiednio małe.